2

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A DATABASE TO SUPPORT
DOD BUSINESS PROCESS REDESIGN

by

William C. Kotheimer Jr.

September, 1992

Thesis Advisor: William J. Haga

Approved for public release; distribution is unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL *(If applicable)* 37	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000	7b. ADDRESS *(City, State, and ZIP Code)* Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(City, State, and ZIP Code)*	10. SOURCE OF FUNDING NUMBERS Program Element No. / Project No. / Task No. / Work Unit Accession Number

11. TITLE *(Include Security Classification)*
A DATABASE TO SUPPORT DOD BUSINESS PROCESS REDESIGN

12. PERSONAL AUTHOR(S) Kotheimer, William, C. Jr.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(year, month, day)*	15. PAGE COUNT
Master's Thesis	From To	September 1992	175

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES

FIELD	GROUP	SUBGROUP

18. SUBJECT TERMS *(continue on reverse if necessary and identify by block number)*
Reengineering, Prototype, Database, CIM

19. ABSTRACT *(continue on reverse if necessary and identify by block number)*

This thesis describes the development of a database to support business process redesign in the Department od Defense (DOD). Business process redesign is rapidly becoming an important part of DOD's Corporate Information Management (CIM) initiatives. DOD is changing the way it does business in order to meet its commitments with fewer resources. In describing the development of a database to support business process redesign, this thesis reveals insights into methods and practices that are changing the way business is practiced. The challange encountered in this project is that the process of business process redesign in DOD is being developed concurrently with the database. In effect, the database is being built to support a process that is itself not fully understood. It was found that sufficient information on business process redesign existed and could be quantified in such a manner as to be made available in a database format. The development of a prototype database progressed to a stage where it could be implemented. The next step is to build a fully functional model of the database in order to evaluate its role in supporting business process redesign.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Haga, William, J	22b. TELEPHONE *(Include Area code)* (408) 646-3094 22c. OFFICE SYMBOL AS/Hg

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Development of a Prototype Database to Support Business Process Reengineering in the Department of Defense

by

William C. Kotheimer Jr.
Lieutenant, United States Navy
B.M.E., Villanova University, 1985

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1992

Author: William C. Kotheimer Jr.

Approved by: William J. Haga, Thesis Advisor

Kenneth J. Euske, Second Reader

David R. Whipple Jr., Chairman
Department of Administrative Sciences

ABSTRACT

This thesis describes the development of a database to support business process redesign in the Department of Defense (DOD). Business process redesign is rapidly becoming an important part of DOD's Corporate Information Management (CIM) initiatives. DOD is changing the way it does business in order to meet its commitments with fewer resources. In describing the development of a database to support business process redesign, this thesis reveals insights into the methods and practices that are changing the way business is practiced. The challenge encountered in this project is that the *process* of business process redesign in DOD is being developed concurrently with the database. In effect, the database is built to support a process that is itself not fully understood. It was found that sufficient information on business process redesign existed and could be quantified in such a manner as to be made available in a database format. The development of a prototype database progressed to a stage where it could be implemented. The next step is to build a fully functional model of the database in order to evaluate its role in supporting business process redesign.

TABLE OF CONTENTS

I. INTRODUCTION

This thesis describes the development of a database to support business process redesign in the Department of Defense (DOD). As defense budgets shrink and the armed forces are employed in increasingly diverse roles, business process redesign will become an important part of DOD's Corporate Information Management (CIM) initiatives. In effect, DOD must change the way it does business in order to meet its commitments with fewer resources. This thesis will not only describe the design of a database, but will also reveal insights into the business process redesign methods and practices. The challenge encountered in this project is that the *process* of business process redesign in DOD is being developed concurrently with the database.

A. BACKGROUND

Two occurrences in recent years have had significant impacts on the way that the Department of Defense (DOD) operates: (1) Congress has become increasingly displeased with the way DOD manages its information technology and (2) the end of the Cold War has lead to a down-sizing of the defense establishment.

1. Creation of the CIM office

In July 1989, the House Armed Services Committee, responding to GAO reports of mismanagement of automated data processing in DOD, suggested that funding for DOD investments in information technology would no longer be forthcoming until the department established a unified, non-duplicative, comprehensive strategy for its

Information Systems (IS). At the time, DOD was spending nine billion dollars annually on IS resources. In response to Congress's suggestion, the Secretary of Defense appointed a Deputy Secretary (DSD) to manage the DOD comptroller office, which included the office of DOD Information Resources Management (IRM). The DSD brought the corporate information management (CIM) strategy to the office, devised to bring information resources together across divisional boundaries. In November 1989, the CIM office was created under the DOD deputy comptroller for IRM. A director was appointed who began implementing the DSD's strategy with an emphasis on unification and standardization of information resources.

2. CIM Initiatives

In October 1990, the Senate took one billion dollars out of the IS request in the Defense Appropriations Bill and allocated it to the CIM office so they could begin implementation of CIM initiates. The bulk of this funding would be returned to the services and agencies from which it was taken, but only if the systems that they sought to fund met CIM standards. The message sent from Capitol Hill was that, from then on, proposals for IS must possess the capability for DOD-wide integration and standardization. In December, 1990, the Secretary of Defense moved the CIM office from the comptroller office and placed it within the domain of the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence (ASD(C^3I)). In January 1991 (ASDC^3I) created the position of Director of Defense Information (DDI). An IS executive of national repute was appointed to the post. Within six months, the DDI expanded the CIM concept to encompass business process redesign.

3. Process Redesign Research

Events in the former Soviet Union between August and December 1991 ultimately lead to the disintegration of the USSR and effectively brought an end to the Cold War. In light of the diminished threat to national security, significant cuts were made to DOD's budget with more and deeper cuts expected in the future. Rather than making across-the-board cuts in information systems, the DDI sought to carve non-value-added activities out of business processes. The DDI's message was that if DOD was going to be smaller, then it was going to have to work smarter. Only after a process was redesigned to effectively incorporate the benefits inherent in information systems would it be considered a candidate for automation. In January 1992, faculty and students of the Department of Administrative Sciences of the Naval Postgraduate School (NPS) undertook a research project aimed at facilitating business process redesign. Specifically, an NPS faculty-student team would model the "how" of business process redesign using the IDEF modeling tool. The resultant model would be incorporated into a process redesign guide book for DOD functional managers. As a supplement to this guide book, it was anticipated that some type of database of business redesign methods, best business practices and redesign experts should be developed. In March 1992 the research team participated in a five-day IDEF modeling workshop aimed at describing the activity structure for describing how to redesign business processes. The team designated themselves as the Re-design Expert And Practices (REAP) working group. As expected, a business redesign database was determined to be a necessary companion to the redesign guide book. This database was designated the REAP database.

II. DEVELOPMENT METHODOLOGY

A. DEVELOPMENT CONCEPT OVERVIEW

Before beginning a software development project it is necessary to decide upon a software engineering paradigm to follow. In the case of the REAP database, the information system development process suggested by the project sponsor dictates a specific paradigm be used. This development concept was developed by the U.S. Army Corps of Engineers (Figure 1). It consists of four phases:

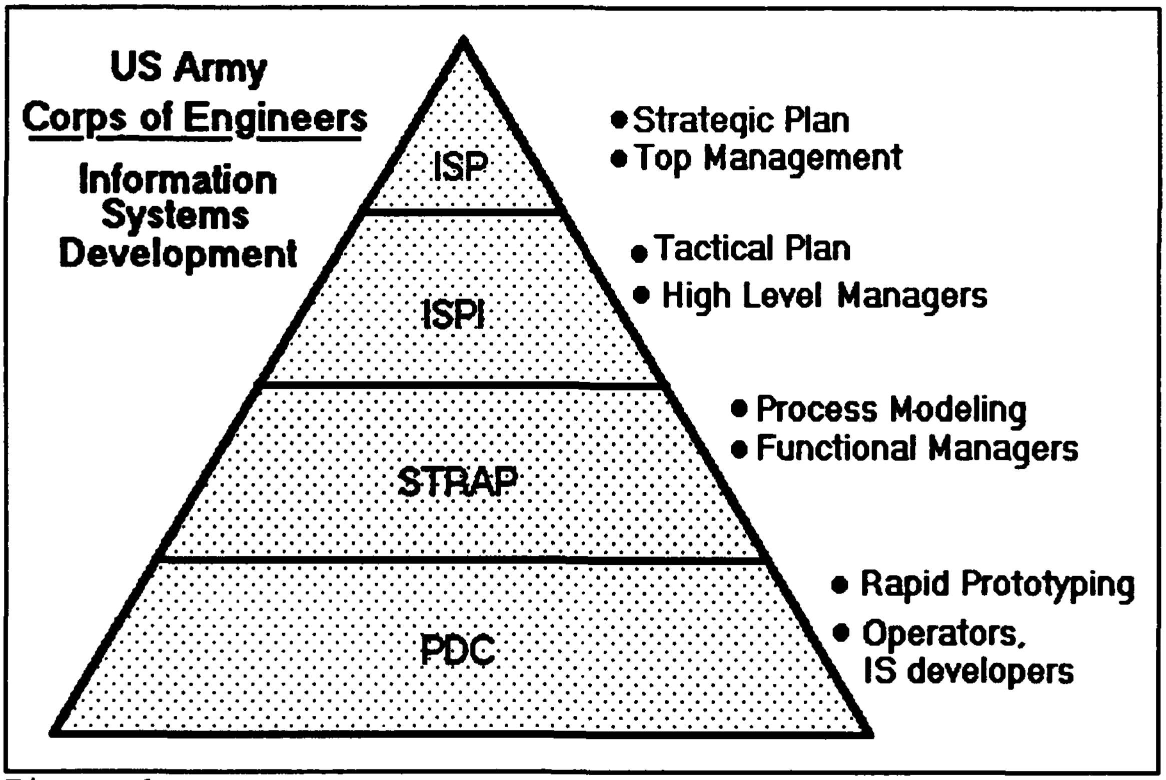


Figure 1

Phase 1. - ISP: Information Systems Planning

The strategic plan: Top managers plan what is needed from the information systems; what the system is to do.

Phase 2. - ISPI: ISP Implementation

The tactical plan: Managers at the next lower level take the ISP plan and implement it by defining architecture, management structure and project slates.

Phase 3. - STRAP: STructured Requirements Analysis Plan

Process modeling: Functional managers, (the information users and system operators) go off site for four weeks and define the current business process. Using IDEF tools, they model the activities of the business process and data elements of business rules. One month later the same group develops a model of the future business process using IDEF.

Phase 4. - PDC: Prototype Development Concept

System development: A group of operators, intimately familiar with one of the processes modeled during the STRAP phase, meet with Information System (IS) developers and create a system using rapid prototyping methodology. The foundation of their work is the model of the future

process developed during the STRAP. This phase is allowed six to nine months to produce a working system. [Spivey, 1992]

The first two phases of this process were conducted at the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence (ASD(C^3I)) and Director of Defense Information (DDI) levels. The March 1992 REAP workshop served as a modified STRAP phase[1]. The remaining phase is the subject of this thesis. The paradigm for the development of the REAP database is software prototyping.

B. THE PROTOTYPING PARADIGM

Prototyping is a process where working models, functionally equivalent to subsets of the target system, are built by the software developers and demonstrated to the end users. The prototype paradigm, as defined by Pressman [1992], is illustrated in Figure 2. The paradigm begins with gathering and refining the system requirements. During this phase, "the information domain and the functional and behavioral domains of the problem" [Pressman,1992] are represented. A quick design is produced based on the system requirements. A prototype model is built (coded) and tested. This working model is evaluated by the customer/end user. The results of the evaluation are used to refine the prototype and a new design is quickly produced. This iterative process occurs until the prototype meets the customer's requirements. [Pressman, 1992] Once a prototype model is accepted by the customer, the complete system is rebuilt, keeping all

[1] Since there was no existing process improvement process to model, the focus of the STRAP was solely on the model of the future system.

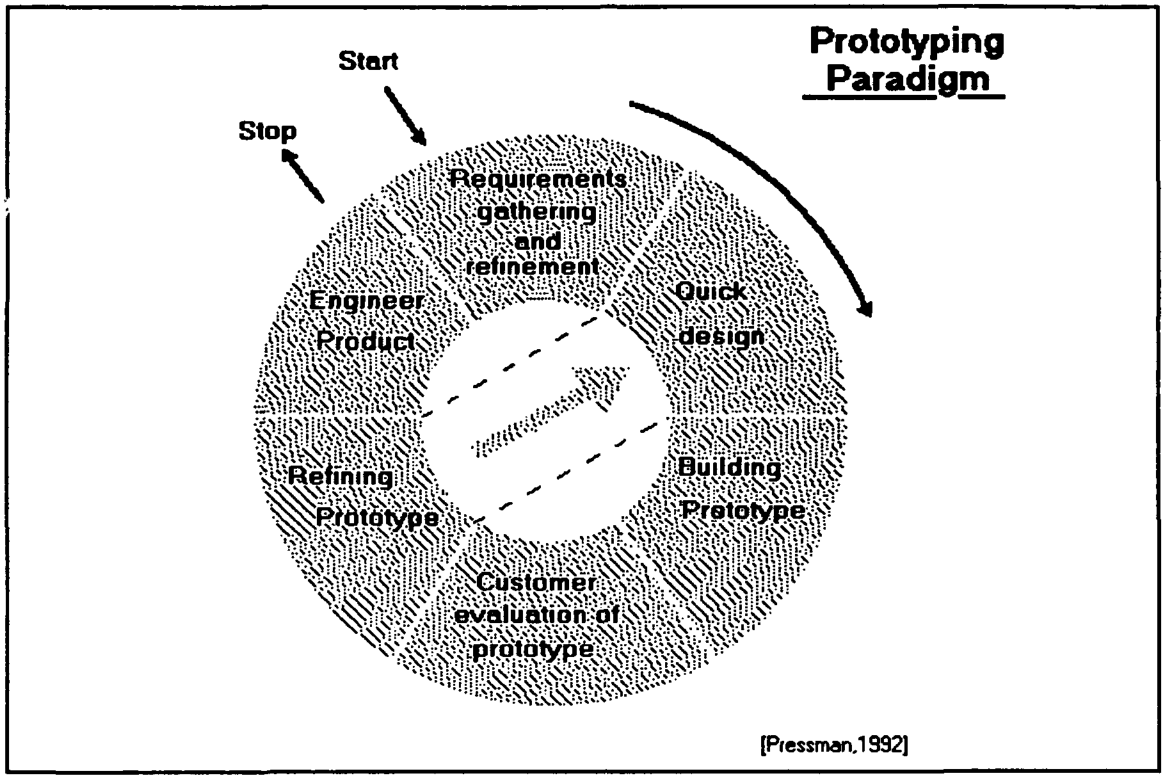


Figure 2

the attributes of the accepted prototype. This is done because, as Brooks [1975] states, "The developer often makes implementation compromises in order to get a prototype working quickly" The purpose of this last phase is to produce a system that is more structured and maintainable than the prototype.

C. SPECIAL ISSUES INVOLVED IN THE REAP DATABASE DEVELOPMENT

Development of the REAP database would be shaped by two special issues not normally experienced in typical software projects.

1. A different means for defining system requirements

Databases are normally developed to support fairly well understood operations (such as inventory control or customer orders) and easily identifiable end users (such as

warehouse managers or customer service personnel). This means that system requirements can be fairly well defined by modeling the process which the system is to support and defining relationships between the system's data. The REAP database was conceived as a support to a process (business redesign in DOD) that was itself under development. Additionally, business practices redesign is not an everyday business function and has been practiced by not more than a handful of DOD organizations. This meant that the requirements could not be produced by database's intended end users; DOD functional managers conducting business process re-design. Instead, the system requirements were developed by the REAP project team at NPS as part of the process improvement process model.

2. Split development effort

Under the Corps of Engineer's concept, the end users and IS developers work together to produce the desired system. The operators contribute their understanding of the process to be supported while the IS developers bring their expertise in technology, programming and system architecture to the project. Working together allows for the easy exchange of ideas and concerns and normally results in rapid development of the target system. A key element of this cooperation is the transmission of a clear understanding of the system's requirements. Pressman [1992] highlights the importance of good, unambiguous requirements when he states, "Requirements analysis is the first technical step in the software engineering process. It is at this point that a general statement of software scope is refined into a concrete specification that becomes the foundation for the software engineering activities to follow." Unfortunately, time and

geography did not permit the NPS REAP team and the DDI programmers to work as a group on the REAP database. The agreement between the NPS Department of Administrative Sciences and the Office of the Director of Defense Information (DDI) states that the REAP project team is to determine the "scope, configuration, architecture, ownership and maintenance" [Department of Administrative Sciences Letter, Feb 1992] of the REAP database. Through informal agreement it was understood that DDI programmers would take NPS prototype and use it to produce the deployed REAP database. With the development effort thus split, it is important that, as the *de facto* end users of the REAP database, the NPS REAP team effectively communicate the system requirements to the DDI programmers. The focus of the NPS produced prototype is the "information, functional, and behavioral domains of the problem" [Pressman, 1992]. The DDI programmers will address the technical details of the system in order to produce a quality and maintainable product.

D. REAP DATABASE PROTOTYPE ATTRIBUTES

The prototype database developed is a work-along model, intended to be as functional as possible to the DDI programmers. Its goals are to convey data definitions and relations, query definitions and display formats to the DDI programmers. The functionality of the prototype is limited to data query and display mechanisms. It is reasoned that data entry, update, verification and deletion mechanisms could best be designed and implemented by DDI programmers.

E. PROTOTYPE DATABASE DEVELOPMENT PROCESS

To effectively communicate the user's requirements, the REAP database prototype needed to be sufficiently and clearly documented so as to be understandable to the target system developers. A formal database management system (DBMS) development model was chosen with minor adjustments, conforming to the prototyping paradigm. The model chosen for the development of the REAP database prototype is the process for database development outlined by Kroenke and Dolan [1988]. This process was chosen because it is clear, simple, specific to the development of database applications, and produces cogent documentation that describe the database. This process is broken into five phases:

1. *Definition phase* - The task is clearly defined, the feasibility of a database solution assessed, and the scope of the database delineated.

2. *Requirements phase* - Data requirements are determined and update, display and control mechanisms are described. The products of this phase include data object diagrams and data flow diagrams.

3. *Evaluation phase* - Possible system architectures are identified and evaluated as to how well they meet the database application requirements. The system architecture that best meets these needs is chosen as the platform for the database.

4. *Design phase* - Database design and application design are formulated. The database design includes file and record structures and relationships. The application design includes data views, display and control mechanism specifications, and program logic.

5. *Implementation phase* - The operational (prototype) database and appropriate documentation are produced. This phase includes writing and debugging the application and populating the database. [Kroenke and Dolan, 1988]

To fit Pressman's prototyping paradigm, Kroenke and Dolan's *definition, requirements* and *evaluation* phases, taken together, make up Pressman's *Requirements gathering and refinement* phase, while Kroenke and Dolan's *Design* and *Implementation* phases will correspond to Pressman's *Quick design* and *Implementation* phases, respectively. At the end of each phase, a phase report, including diagrams, definitions, program logic and/or source coding specific to the phase, are produced. The prototype to be produced by the NPS REAP project is a compilation of these reports and the prototype application and data, on floppy disk. The prototype is sent to the Office of the DDI for the *Customer evaluation* and *Prototype refining* phases. The DDI programmers and/or developers should then take the REAP database prototype through at least one more prototyping cycle before engineering the deployable system.

III. DEFINITION PHASE

A. OUTLINE OF DEFINITION PHASE

Kroenke and Dolan [1988] state that the "first phase of an application development project is to define what the project is to do." They break this phase into four parts:

1. Define Task
2. Form project team
3. Establish the scope of project
4. Assess the feasibility of project

B. DEFINE TASK

From 21 to 25 March 1992, three students (Bizell, Kotheimer, and White) three faculty (Euske, Haga and Nevels) and a Dean (Frew) from the Naval Postgraduate School took part in a workshop provided by D. Appleton Inc. The focus of the workshop was to describe a process by which the task of conducting business practices redesign can be learned. The Process Improvement Process (PIP) was chosen as the name for this process. Using the IDEF model of the PIP, part of the group (Haga, Euske and White) began developing a guidebook to assist DOD functional managers in learning and conducting process redesign within their respective organizations. It was determined that a database was needed to supplement the guidebook. The argument was made that in order for the guidebook to be useful for many, different DOD organizations,

it must be generic in nature. Yet each DOD organization has unique characteristics which makes generic guidance less than optimal. Some type of mechanism is needed to provide the information that the guidebook will not (and should not) cover. That mechanism is a database that functional managers can query for re-design information, specific to their type of activity, should fulfill this need. This database was designated the REAP database. Essentially, the REAP workshop served as the STRAP phase for the REAP database development. The task of the REAP database is to provide information that may be specific to a subset of DOD activities on an as needed basis.

C. FORM PROJECT TEAM

The REAP database project team was formed as a subset of the original REAP working group. Haga was to direct the development of the database application based on his experience in the determining the success of new information systems. Euske was to draw on his knowledge of new business, accounting and management practices in order to direct the population of the database. Kotheimer was to develop the REAP database prototype and conduct research in order to obtain populating data.

D. ESTABLISH SCOPE OF PROJECT

The primary function of the full scale REAP database is to provide additional business process redesign information, supplemental to the aforementioned process redesign guidebook. The primary users of the REAP database will be the functional managers and their support staff, engaged in process redesign. The REAP database should allow the user to select and access the information that she/he deems applicable

to their redesign effort. The function of the REAP database prototype is to define the data structure of redesign information and the format by which that information is presented. The redesign information that prototype contains should be as complete as possible given time and resource constraints. The REAP database should be implemented in a medium that will be readily available to the widest possible range of potential users. The content of the information provided by the database should be as comprehensive as possible given its storage medium. As defined during the REAP project workshop/STRAP, this information should include descriptions of applicable redesign analysis, implementation methods and tools, metrics, benchmarks, case studies, accepted Information Technology (IT) solutions, and experts in business process redesign. The presentation of this information should allow the user to easily move between pieces of related material.

E. ASSESS THE FEASIBILITY OF PROJECT

Three elements were addressed in the prototype feasibility assessment: software availability, hardware availability and time constraints.

1. Software Availability

As described by the REAP working group, the REAP database will primarily be a text retrieval database. It is difficult to make a very accurate estimate of the eventual size of the full scale REAP database since the entire project is in its infancy and the amount of information deemed valuable to business process redesign remains

undiscovered. However, based on preliminary literature research, the size of the initial prototype to be delivered in August 1992, was estimated (Table 1).

Table 1. PROTOTYPE REAP DB - INITIAL SIZE ESTIMATION

Data Element	Number of Records	Record Size (English words)
Methods/ IT Solutions	20-30	200
Case studies	10-15	250
Benchmarks/Metrics	10-15	200
Organizations/Experts	10-20	100
Total	50-80	9,500 to 14,750

It was determined that a number of commercial software products (database applications, programming language compilers and text search and retrieval programs) existed that could handle a database of this size. These products ranged in price from approximately three hundred dollars to several thousand dollars. Most of these applications were well within the research budget or available under site license at NPS.

2. Hardware Availability

The hardware required to run the aforementioned products ranged from desktop personal computers (IBM PCs and Macintosh systems) to workstations

(SUN/Sparc workstations) and mainframe computers (IBM/AMDAHL). Access to each of these machines is available at NPS.

3. Time Constraints

At the time that the definition phase was conducted, there were 21 weeks before the REAP database prototype was due (first week of September, 1992). It was determined that in that time it would be feasible to design and build a prototype with most, if not all, of the main query and display functions of the deployment version REAP database.

IV. OVERVIEW AND OBJECTIVES OF THE REQUIREMENTS PHASE

A. IMPORTANCE OF THE REQUIREMENTS PHASE

The requirements phase of software development is especially critical. Pressman [1992] states, "A complete understanding of software requirements is essential to the success of a software development effort. No matter how well designed or well coded, a poorly analyzed and specified program will disappoint the user and bring grief to the developer." The products of the requirements phase are the requirements specifications which are the basis for the software application design. Incomplete, misinterpreted, or unrealistic requirements are many times the root cause of software project failures. Additionally, frequent changes in the requirements specifications occurring after the requirements phase has been completed can cause projects to slip behind schedule. Boehm states,

> "Current approaches to the software process make it too easy for software projects to make high-risk commitments that they will later regret. The sequential, document driven "waterfall" process model tempts people to over promise software capabilities in contractually-binding requirements specifications before they understand the risk implications." [Boehm, 1989]

As evidence, Boehm provides a list of the top ten primary sources of risk in software projects (Table 2), based on a survey of a number of experienced project managers

Table 2. A TOP TEN LIST OF SOFTWARE RISK ITEMS	
Risk Item	**Related to Poor Requirements**
1. Personnel shortfalls	
2. Unrealistic schedules and budgets	
3. Developing the wrong software functions	Yes
4. Developing the wrong user interface	Yes
5. Gold Plating (Extra un-needed features)	Yes
6. Continuing stream of requirements specifications	Yes
7. Shortfalls in externally furnished components	
8. Shortfalls in externally performed tasks	
9. Real-time performance shortfalls	
10. Straining computer science capabilities	

[Boehm, 1989]

Three of the top ten risk sources (3,4,5) can be directly related to poor requirements specifications while a fourth (6) is related to a lack of control over changes made to software requirements. These four items serve to focus attention on two critical issues contended with during the REAP database requirements phase: accuracy and completeness. Accuracy means that the requirements specifications produced need to clearly describe the correct software functions and the correct user interfaces while excluding unnecessary functions. Completeness is important since the REAP database requirements are to be used, not only by the REAP database team in the design phase, but also by the Director of Defense Information's (DDI) programmers when

implementing the full scale system. Pressman [1992] states that the "genesis of most new systems begins with a rather nebulous concept of desired function... the system engineer must bound the system by identifying the scope of function and performance that are desired." The objective of the requirements phase is to produce an accurate, comprehensive representation the scope, function and performance of the REAP database.

B. OUTLINE OF REQUIREMENTS PHASE

According to Kroenke and Dolan [1988], the purpose of the requirements phase is two fold:

1. Identify data objects and define their structure.
2. Determine the functional components of the database.

The focus of this analysis is on what is needed and why it is needed, not how it will be implemented.

Identifying and defining data objects is fairlystraight forward. The PIP model developed by the REAP working group during the STRAP phase described the kind of information that the database was to provide. These descriptions are analyzed and expanded to form the data objects specifications.

Defining the functional components of the system is more complicated than defining the data elements. While general functional requirements were discussed during the STRAP phase, other factors, such as the amount and type of information to be presented and data relations, must all be considered during the functional components requirements analysis.

C. METHODS FOR DATA REQUIREMENTS SPECIFICATION

Kroenke and Dolan [1988] call for the development of "data objects" to formally specify data requirements. They describe a data object as a "named collection of properties that sufficiently describes an entity in the user's work environment". Pressman [1992] states, "Data objects are related to one another [and that] the relationships are always defined by the context of the problem that is being analyzed." While Kroenke and Dolan's definition is a good starting point to begin building data objects, Pressman provides a better sense of what the data objects should provide. If analyzed and specified correctly, the data objects will not only provide a sufficient description of the entities in the user's work environment (in this case, sufficient information on some aspect of process re-design), but will also define the relationships between data objects, in the context that the user "sees" those relationships. In other words, the data specifications should be a description of the user's environment from the user's perspective.

When describing a data object, the term property "represents a characteristic of the corresponding entity that is important to one or more users of the database application" [Kroenke and Dolan, 1988]. A property can be information "contained" in the data object or it can be other data objects. For example, a data object called Ship may contain the property Sailor, which would itself be a data object containing all the properties needed to describe a sailor. In general the domain of a property is the set of all possible values the property can have. The domain of a non-object property consists of semantic

description (describes the function or purpose) and the physical description (indicates data type). For example when defining the property Ship Name as "Text, 20 characters; the name of a U.S. Navy ship", the first half (Text, 20 characters) is the physical description and the latter (name of a U.S. Navy Ship) is the semantic description. The domain of an object property is a set or subset of object instances which represent a sufficient "view" of the object. Using the example of Ship and Sailor: The Sailor object may contain the properties name, rank, rate, blood type, and ssn. While a property of the Ship object, the Sailor object may represent only name and rank. Data objects that contain other data objects are known as compound objects [Kroenke and Dolan, 1988].

Accurate and comprehensive specification of compound objects is critical because they define data relationship structure in the database [Kroenke and Dolan, 1988]. These relationships are critical to the success of the REAP database as they are what will make the REAP database useful to a DOD functional manager. For example, by relating business analysis methods with organization case studies and business analysis experts, the user can study the fundamentals of a specific method, review related case studies to see how it has been implemented in other organizations, and find someone with expertise in the area who can aid in its implementation.

D. METHODS FOR DEFINING FUNCTIONAL REQUIREMENTS

Kroenke and Dolan [1988] break the specification of a database's functional requirements into two steps: the identification of applications and then the determination

of the functional components of each application. Before one can identify an application, one must define what an application is.

All information systems process data to produce information and maintain sto::ed data. [Whitten, Bently and Barrow, 1989]. The mechanisms by which data is processed are called applications. In essence, applications are the interface between the data in the database and the user.[Kroenke and Dolan, 1988] It can be argued that database applications are the devices that transform data into the information. Specifically, database applications receive instructions from the user, locate and retrieve the applicable data, if necessary combine it with related data, and present it in a form that "makes sense" to the user.

With the concept of a database application thus defined, it can be seen that the objective of the functional requirements is to identify the applications of the REAP database and specify what will be required of them.

1. Data Flow Diagrams

The first step in this process will be to define and analyze the functionality required of the REAP database in order to determine the number of applications needed and determine their individual functions. Essentially, the processes that are performed in the course of the database's operations are to be identified and analyzed and modeled. This modeling will take the form of the construction of data flow diagrams. Page-Jones [1989] states that the "data flow diagram (DFD) is used to partition a system, and it is chiefly this tool that the structured specification owes its desirable qualities of being graphic, concise and partitioned. Simply stated, a data flow diagram "shows the active

components of system and the data interfaces between them." Most DFDs are constructed by simply analyzing and modeling the data flows in an existing system (either automated or manual). Since there is no existing system to model, the DFDs for the REAP database are created based on the initial functional requirements described in the STRAP phase and logical assumptions about user requirements, drawn from the relations between the data objects. In this case, the partitioning and concise nature of the DFD tool is used to bring definition to the functions of the REAP database.

Data flow diagrams are generally considered to have four components, external entities, data flows, processes and data stores. Pressman [1992] provides useful, succinct definitions of these components:

External Entity A rectangle: A producer or consumer of information that resides outside of the bounds of the system to be modeled.

Data Flow An arrow: A data item or collection of data items; the arrow head indicates the direction of flow.

Process A circle or oval: A transformer of information that resides within the bounds of the system to be modeled.

Data Store	Two parallel lines: A repository of data that is to be stored for use by one or more processes; it may be as simple as a buffer or que or as sophisticated as a relational database.

2. Determining Functional Components

Specifying the system applications with data flow diagrams is not enough. Kroenke and Dolan [1988] call for the determination of the functional components of each application. Specifically, the update, display and control mechanisms for each application are defined. It was stated during the definition phase that the focus of the REAP database prototype developed by NPS would be limited to the display mechanisms. In keeping with Kroenke and Dolan [1988] process, the specifications of the display mechanisms consist of the identification of the data objects processed and descriptions of the display mechanisms, to include output description, source data, processing notes, and estimated volume and frequency of use.

The summation of these products is considered the specification of the functional requirements and should prove sufficient basis for the REAP database application design.

V. DATA REQUIREMENTS ANALYSIS AND SPECIFICATION

A. INITIAL DATA REQUIREMENTS SPECIFICATIONS

The product of the STRAP phase was the Process Improvement Process (PIP) activity, Create a Methodology for Process Redesign, designed using IDEF modeling techniques (Figure 3). User requirements for the REAP DB were discussed and outlined during the modeling of this activity. In general, IDEF modeling depicts an activity by describing its inputs, controls, outputs and mechanisms. The IDEF model produced during the workshop for the PIP activity Create a Methodology for Process Redesign consisted of five sub-activities. The REAP DB was identified as a mechanism for four of these activities.

During the STRAP phase, the REAP working group, determined that the REAP database would contain the following information:

1. Lists of names and contact points for experts and facilitators in activity redesign methods and techniques.
2. Lists and brief descriptions of methods and techniques for modeling, portraying and analyzing existing business processes.
3. Lists of activities in DOD and firms in the private sector that have already experienced process redesign and offer contact points willing to share their experience.

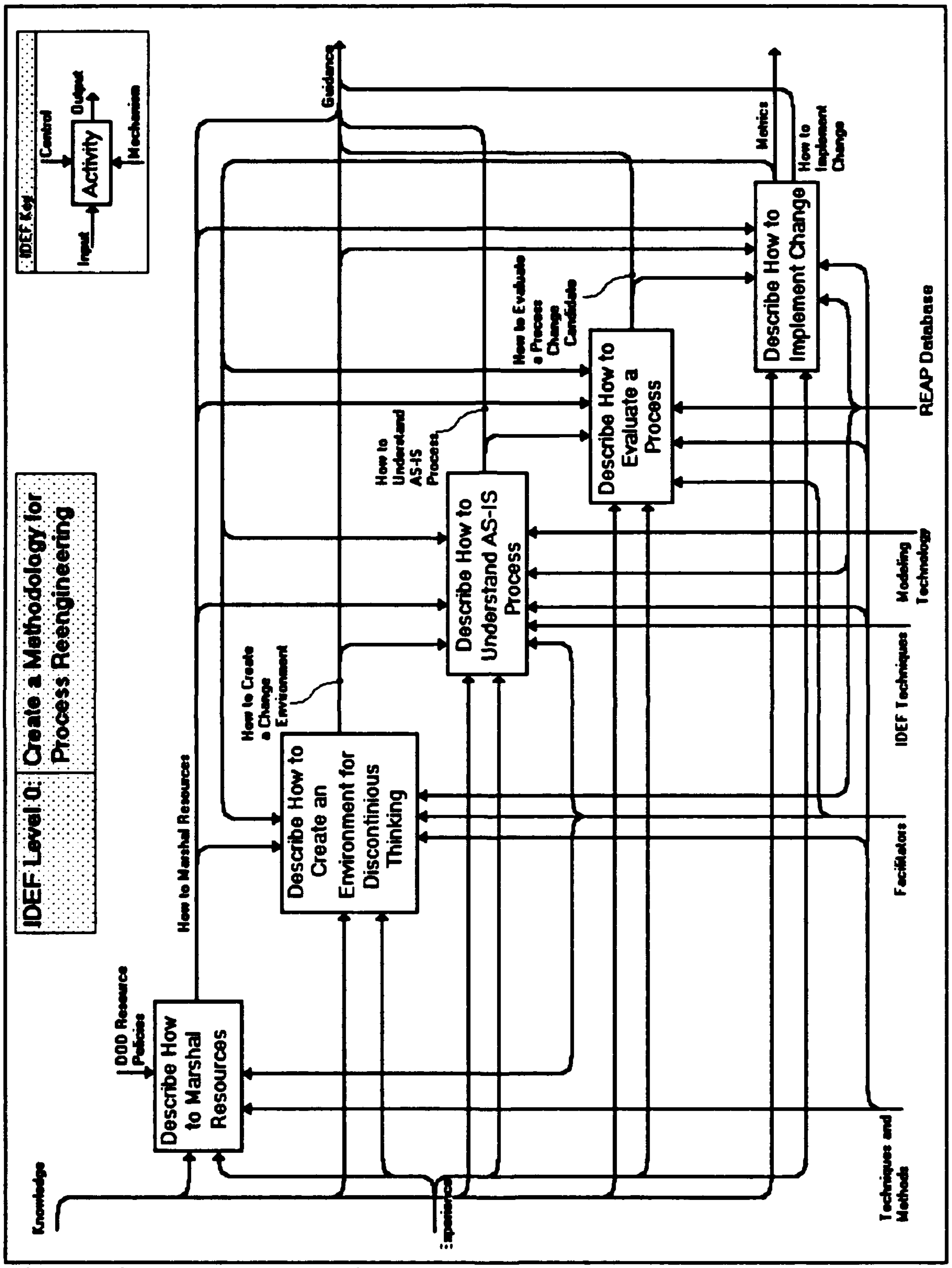


Figure 3: IDEF Level 0 model

Of the four activities identified, one (Describe how to Create an Environment for Discontinuous Thinking) was determined to need a database mechanism to provide generic redesign information unrelated to functional area, and the other three activities (Describe how to Understand Process, Describe how to Evaluate a Process, Describe how to Implement Change) were determined to need a database mechanism that would provide information specific to a manager's functional area. Generic redesign support was determined to be a list and explanation of "change environment" methods, experts and organizations. Support to specific functional areas was determined to be lists and descriptions of process analysis methods applicable to the functional area as well as similar DOD or private organizations that have experienced process improvement.

B. REFINEMENT OF DATA REQUIREMENTS

The data requirements defined during the STRAP phase are not comprehensive enough to be used as the basis of a database design. The process is to take each of the three broad data specifications produced by the STRAP phase and refine them into data objects. These data objects are represented graphically by means of object diagrams. Additionally, the object properties are defined in object specifications.

1. Experts and Facilitators

The first data specification, lists of names and contact points for experts and facilitators in activity redesign methods and techniques, can be broken into two data objects, an object called Expert and an object called Organization. The Expert object is

fairly simple, containing properties that sufficiently identify the expert (Expert name, salutation, position) and provide contact information (address, telephone number).

The meaning of the term "facilitator" in the STRAP phase definition is a bit ambiguous. It can mean a person who facilitates an activity redesign method or technique or a consulting or education organization that provides redesign services. Since the data object Expert could be used for individual facilitators, we can define facilitator as an organization that provides education, consulting and/or facilitating services in activity redesign. The Organization object is created and given identification, description and contact information properties (Organization name, Org. description, Org. address, Org. phone) as well as properties that describe the services or products the organization provides (Org. product). There may be instances where an expert is employed by a consulting organization. A relation between the Organization and Expert objects is built on this possibility. The Organization object is added as a property of the Expert object and the Expert object is added as a property of the Organization object, linking Expert object instances with corresponding Organization object instances.

2. Analysis and Redesign Methods and Techniques

The second data specification, lists and brief descriptions of methods and techniques for modeling, portraying and analyzing existing business processes, needs much refinement. A data object is created called Method that contains properties that identify, describe and summarize the benefits of a business analysis or redesign method (Method name, Summary, Method results). A relation between the Method object and the Expert object and a relation between the Method object and the Organization object

are created to link experts and consulting organizations that specialize in particular analysis methods. Both the Expert and Organization objects are added as properties of the Method object and the Method object is added as a property of the Expert and Organization objects. Since it is logical to assume that a user would search for a particular method before looking for experts or organizations that practice the method, queries should be structured so as to only allow finding experts or organizations based on method.

During research into new business analysis and redesign methods and techniques it was discovered that most are sufficiently complicated that the "brief description" called for in the STRAP requirements cannot give more than a general overview of the process. During a literature search for REAP material, it was discovered that there are a great number of books, articles, and papers, available from a variety of sources, which provide comprehensive descriptions and discussions of many of the more popular business analysis and redesign methods. A data object called Publication is created to capture this information. The properties of the Publication object include identifiers (Title, Publisher and Year published) as well as summary of the publication (Pub summary). Since the author(s) of a publication can be considered an expert(s) in the method(s) described, a relation is created between the Publication object and the Expert object. Any given Method instance may have more than one corresponding Publication instance related to it and any given publication instance may cover multiple methods. This possible many-to-many relation is described by making the Method object a multi-valued property of the Publication object and the Publication

object a multi-valued property of the Method object. This means that when the user is viewing a description of a particular method she/he will be able to call up lists and summaries of publications that discuss the method and when he/she is viewing a summary of a publication, he/she will be able to call up lists and descriptions of other methods that the publication may discuss.

Many articles and texts reviewed used case studies to illustrate business analysis or redesign methods in practice. It was determined that the case study format would be useful to REAP database users by providing practical information on these methods. A Case Study object is created with properties to identify and describe the case (Case Name, Case summary). To provide information on the subject of the case study, the Organization object can be used and is added as a property of the Case Study object and Case Study added as a multi-valued property of the Organization object. The Case Study object is added as a multi-valued property of the Method object.

The point was raised during the STRAP phase that some business analysis techniques are aided by the use of computer applications. The computer based modeling application used to produce the IDEF model of the PIP was used as an example. A data object called Software with identification and descriptive properties (Application name, Software (S/W) description, Hardware (H/W) requirements, Operating systems) is created to describe these applications. To establish the relation between a specific Method and software applications that can be used to support it, the Software object is added as a multi-valued property of the Method object and Method is added as a multi-valued property of the Software object. A property representing the company that

produces the application is added (S/W Publisher) in case the user wants to buy copies of the application.

3. DOD activities and private firms with redesign experience

The third data specification, Lists of activities in DOD and firms in the private sector that have already experienced process redesign and offer contact points willing to share their experience, appears fairly straightforward. The Organization data object is sufficient to identify and describe any DOD activity or private firm that has experienced process redesign.

But what organizations should be included in the REAP database? Obviously, not every DOD activity or private firm that conducts process redesign completes the process successfully. Ideally, only the success stories, or well documented failures should be included. But to make a list of these organizations useful, some type of description of how an organization went about its process reengineering, what the new process is, and how the new process worked or failed is needed. It is these two ideas, listing only the organizations with significant redesign efforts and providing descriptions of their new processes, that brings up the business benchmarking concept. Benchmarking is "the continuous process of measuring products, services, and practices against the toughest competitors or those companies recognized as industry leaders (David T. Kearns, CEO, Xerox Corporation)" [Camp, 1989]. Through benchmarking, an organization will discover and employ increasingly better business practices until ideally they become the benchmark themselves. The underlying theme of benchmarking is continuous process evaluation, comparison, and if necessary change.

A Benchmark data object is obviously required. But before the properties of this object can be described, the information that benchmarking produces must identified and defined. The Benchmark object will need an identification property (Benchmark name), and a property identifying the benchmark organization (the Organization data object). The identity and format of the other properties are defined through research into the Benchmarking. Camp [1989] states that "Benchmarking is not just an investigation of the metrics of external business functions, but an investigation to determine what practices are being used to ensure effectiveness...and which practices achieve the metrics" It is important that a property of the Benchmark data object be a description the process(s) involved in achieving benchmark results (Process summary).

While Camp implies that metrics are of lesser importance, they are necessary to compare a benchmark process against other processes. Current literature indicates that business metrics should provide "information on how the work is currently being performed, whether it contributes to the corporate objectives, what the drivers of activities are, and how the system facilitates behavioral incentives to improve effectiveness." [Brimson, 1991] The conclusion can be drawn that a description of the metrics used to measure a benchmark process would aid in the understanding of that process. To facilitate this, a separate data object called Metric is created with properties to identify a business metric (Metric name), describe its uses (Use) and describe its means of measurement (Units). The Metric object is added as a multi-valued property of the Benchmark object along with a multi-valued property (Value) to represent the measure of a benchmark process (in the corresponding metric's units). For example,

given a benchmark process for preparing a voucher that takes 3.5 man-hours, the man-hour is the metric and 3.5 is the value. Benchmark is added as a multi-valued property of the Metric object since one metric may be used to measure several benchmark processes.

All together, the above properties give a fairly complete representation of a benchmark process. However a literature search for examples of business benchmarks revealed that there are often features of a benchmark process and/or organization that make copying the process impractical or impossible. Card [1991] warns of a "cargo-cult mentality"[2] approach towards benchmarking that can happen when people try to copy a successful process without understanding the basis on which it was formed. The lesson taken from this is that more than just a description of the benchmark process, its metrics and its organization is needed. An explanation of why and how a benchmark process came to be implemented is also necessary. A summary or case study of the redesign process that lead to a benchmark process should provide sufficient explanation of these "how's" and "why's" and would maintain the focus of this part of the REAP database, namely organizations that have undergone process redesign. The Case Study object can be pressed into service in order to provide summaries of these successful redesign cases. The Case Study object is added as a single value property of the Benchmark object and Benchmark is added as a single value property of Case Study. The user's view of the

[2] The cargo cult was a group of natives, living in the South Pacific during the late 19th and early 20th century. After observing that valuable cargos regularly arrived at harbors and airports, they gave up farming and fishing to build mock ports and airfields in the vain hope of attracting planes and ships bearing cargo.[Card, 1991]

Benchmark object now includes a description of the process, a measure of the effectiveness of the process, a description of the metric(s) used to measure the process, a description of the benchmark organization and a summary of how and why the process was implemented. This is the complete set of information that is deemed necessary to provide an understanding of a benchmark.

4. Information Technology (IT) Solutions in Process Redesign

A literature search for business process reengineering success stories was conducted as part of the feasibility analysis for the REAP database. A common element found in almost all of the cases was the introduction of information technology to automate or enhance some aspect of the process that had previously been performed by a human being. Examples of information technology being introduced as part of a process improvement effort include wide area or local area networks (WANs or LANs respectively), computer graphics and drafting, computer aided manufacturing, document imaging and electronic storage, electronic signature, database management systems, decision support systems and expert systems. While this aspect of the process improvement process was not formally specified during the STRAP phase, it is deemed important enough to be represented in some way in the REAP database. A data object called IT Solution is created with properties to identify the solution (Solution name), describe the technology (IT summary), specify the system requirements (Sys requirements), and describe the results that can be achieved by its implementation (IT impact). In order to provide real examples of the introduction of the technology, the Case Study object will be used and is added as a multi-valued property of the IT Solution

object and the IT Solution object is added as a multi-valued property of the Case Study object. In order to provide information on the software applications being used in the IT solution, the Software object is added as a multi-valued property of the IT Solution object and the IT Solution object is added as a multi-valued property of the Software object.

5. Relating CIM Areas to the database Objects

While not formally defined during the STRAP phase, it was discussed and understood that in order for the REAP database to be useful, the user would have to be provided with some way of filtering out the information that is not applicable to his/her functional area. In order to accomplish this, a final data object called Area is needed. But before we can create the Area object we must be able to describe a functional area. It is understood that, *combined, all of the DOD functional* areas cover every business aspect of DOD. The question is where to draw the line between instances. Too narrow a definition of functional area will result in too many Area object instances for the user to choose from and too much information filtered out. On the other hand, too broad a definition will result in too little information being filtered out. While the exact division of functional areas need not be addressed during the requirements phase, the requirements for the Area object must be specified so that there is latitude for refinement of the definition of functional area. Therefore it was decided that, at least initially, functional areas be defined as the Corporate Information Management (CIM) areas as described by Strassmann [1992] (Table 3). The Area object is specified so as to be capable of providing sufficient descriptions of these areas.

Table 3.
DOD CORPORATE INFORMATION MANAGEMENT (CIM) AREAS

CIM Area	Responsible Organization
Civilian Payroll	Financial & Accounting Services
Travel	Financial & Accounting Services
Retired Pay	Financial & Accounting Services
Contract Payment	Financial & Accounting Services
Financial Operations	Financial & Accounting Services
Government Furnished Materials	Financial & Accounting Services
Civilian Personnel	Air Force
Depot Maintenance	Air Force
Materials Requirements	Air Force
Distribution Center Operations	Defense Logistics Agency
Materials Asset Management	Army
Technical Documentation	Army
Materials Item Introduction	Marine Corps
Materials Acquisition Management	Navy
Engineering Drawing Management	Navy
Composite Health Care System	Medical Services
Blood Management System	Medical Services
Medical Logistics, Dental Services	Medical Services
Command and Control	Joint Chiefs of Staff

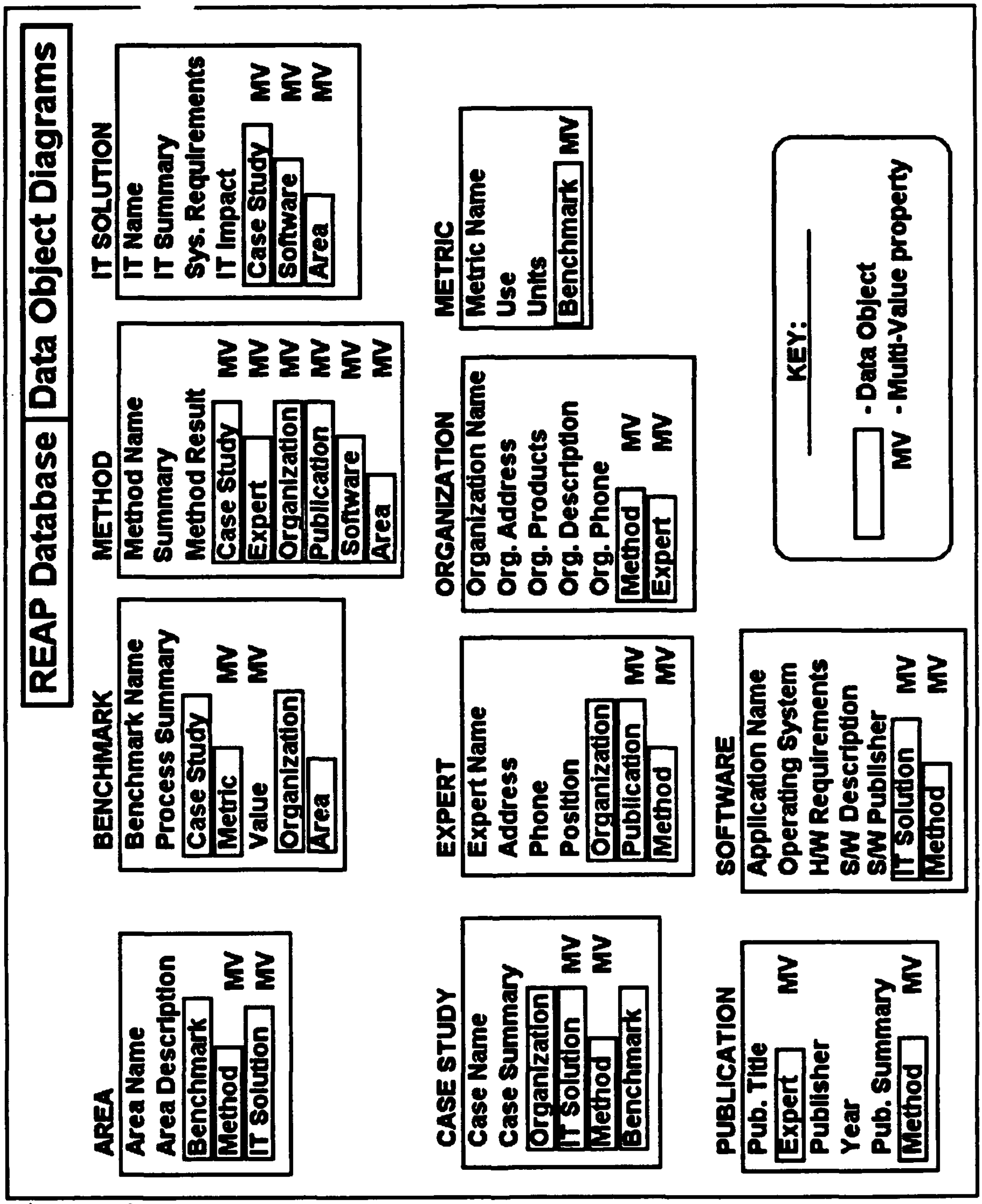
The CIM areas were determined to be good categories for dividing and relating information in the REAP database. Should more refinement be needed, these instances can be broken down into smaller categories, e.g. Travel could be broken into

travel authorization, travel claim preparation, and travel funds disbursement. The Area object will have an identification property (Area name) and a descriptive property (Area description). The remaining properties of the Area object are object properties that will establish the relations between the Area object and all the other data objects in the database. The Benchmark object is added as a multi-valued property to provide examples of organizations in the functional area that have achieved benchmark status through process improvement or redesign. At least one benchmark should be sought for each Area object instance. The Method object was added as a multi-valued property to provide information on process analysis and redesign methods. Both generic methods and those specific to a particular functional area can be related. The IT Solution object is added as a multi-valued property to provide information on the impact of information technology in a specific functional area. Thus defined, the Area object is considered to be an association object, an object that documents a relationship between two (or more) other objects [Kroenke and Dolan, 1988]. To enforce these relationships, the Area object is added as an object property to the Benchmark, Method, and IT Solution objects.

C. SUMMARY OF DATA REQUIREMENTS SPECIFICATIONS

In all, ten data objects define the data requirements for the REAP database. All are compound objects, indicating that a relational database will be required to implement the database application. The data object diagrams are shown in Figure 4.

The Benchmark, Method, and IT Solution objects are the primary data objects of the database. All the other objects in the database are related in some fashion to these



REAP Database
Data Object Diagrams
AREA
Area Name
Area Description
Benchmark MV
Method
IT Solution MV
BENCHMARK
Benchmark Name
Process Summary
Case Study MV
Metric MV
Value
Organization
Area
METHOD
Method Name
Summary
Method Result
Case Study MV
Expert MV
Organization MV
Publication MV
Software MV
Area MV
IT SOLUTION
IT Name
IT Summary
Sys. Requirements
IT Impact
Case Study MV
Software MV
Area MV
CASE STUDY
Case Name
Case Summary
Organization MV
IT Solution MV
Method
Benchmark
EXPERT
Expert Name
Address
Phone
Position
Organization MV
Publication MV
Method
ORGANIZATION
Organization Name
Org. Address
Org. Products
Org. Description
Org. Phone
Method MV
Expert MV
METRIC
Metric Name
Use
Units
Benchmark MV
PUBLICATION
Pub. Title
Expert MV
Publisher
Year
Pub. Summary
Method MV
SOFTWARE
Application Name
Operating System
H/W Requirements
S/W Description
S/W Publisher
IT Solution MV
Method MV
KEY:
- Data Object
MV - Multi-Value property

Figure 4

three primary data objects. The Area object acts as the linchpin of the data structure, providing the relation between the three primary objects. In essence the user determines which Benchmark, Method and IT Solution instances they will see when selecting an Area object instance applicable to their functional area.

This arrangement provides a degree of flexibility that should prove beneficial to the REAP database. The user will be able to concentrate on information related to one functional area or review possibly applicable information related to similar areas. A full description of the data objects is listed in Appendix A.

VI. FUNCTIONAL REQUIREMENTS ANALYSIS AND SPECIFICATION

A. INITIAL FUNCTIONAL REQUIREMENTS SPECIFICATIONS

During the STRAP phase, the REAP working group determined that the REAP database would best support the business redesign process if it were capable of providing near immediate responses to user queries. This meant that the REAP database must be an "on line" service. The alternative, sending queries to the operators of the REAP database via e-mail or telephone, and running them as batch processes (with possible turn around times of 24 hours or greater) was deemed unacceptable. It was also determined that, in order for the REAP database to gain acceptance and be considered as a useful tool, the database interface would have to be fairly simple and easy to learn. Users would not be required to learn the database's data structure in order to conduct queries. This meant that a set of built-in, automatic queries are needed to provide for all logical relations between data objects. Using these predefined queries, the database interface should guide the user to the information that she/he is seeking by first soliciting enough information from the user to filter out nonapplicable records and then allowing the user to explore the applicable data in some logical fashion. The application should provide the users with clear choices at decision points and allow them to back out of searches at almost any point.

While not discussed in depth, it was recognized that in addition to end user functions, the REAP database would need mechanisms to add, edit and delete database records. Additionally, it was recognized that some type of mechanism to record usage statistics and produce usage reports may be desirable.

Based on the initial functional requirements, a top level (level 0) data flow diagram of the REAP database application (Figure 5) can be drawn. The diagram reveals that there are actually three separate functions conducted by the database application. The processes Generate Usage Reports and Process Queries are obvious. The combination of Add Records, Edit Records, and Delete Records can be considered as a single function, maintaining the data in the database. The focus of the initial REAP database prototype is the process called Process Queries. Specifications analysis of the Add Record, Edit Record and Delete Record processes is best left for the system developers working in the office of Director of Defense Information (DDI), who have a better understanding of the record maintenance functions and requirements. While the need for a mechanism to compile and report database usage statistics is recognized, this area has not been addressed and is considered beyond the scope of the initial prototype design.

B. DECOMPOSITION OF INITIAL FUNCTIONAL REQUIREMENTS

Since the focus of the REAP database prototype is the user query and information display, special attention will be paid to the analysis of how the user should be able to control the queries, and how queries and displays should be coordinated to provide the user with the right information at the right points in the search.

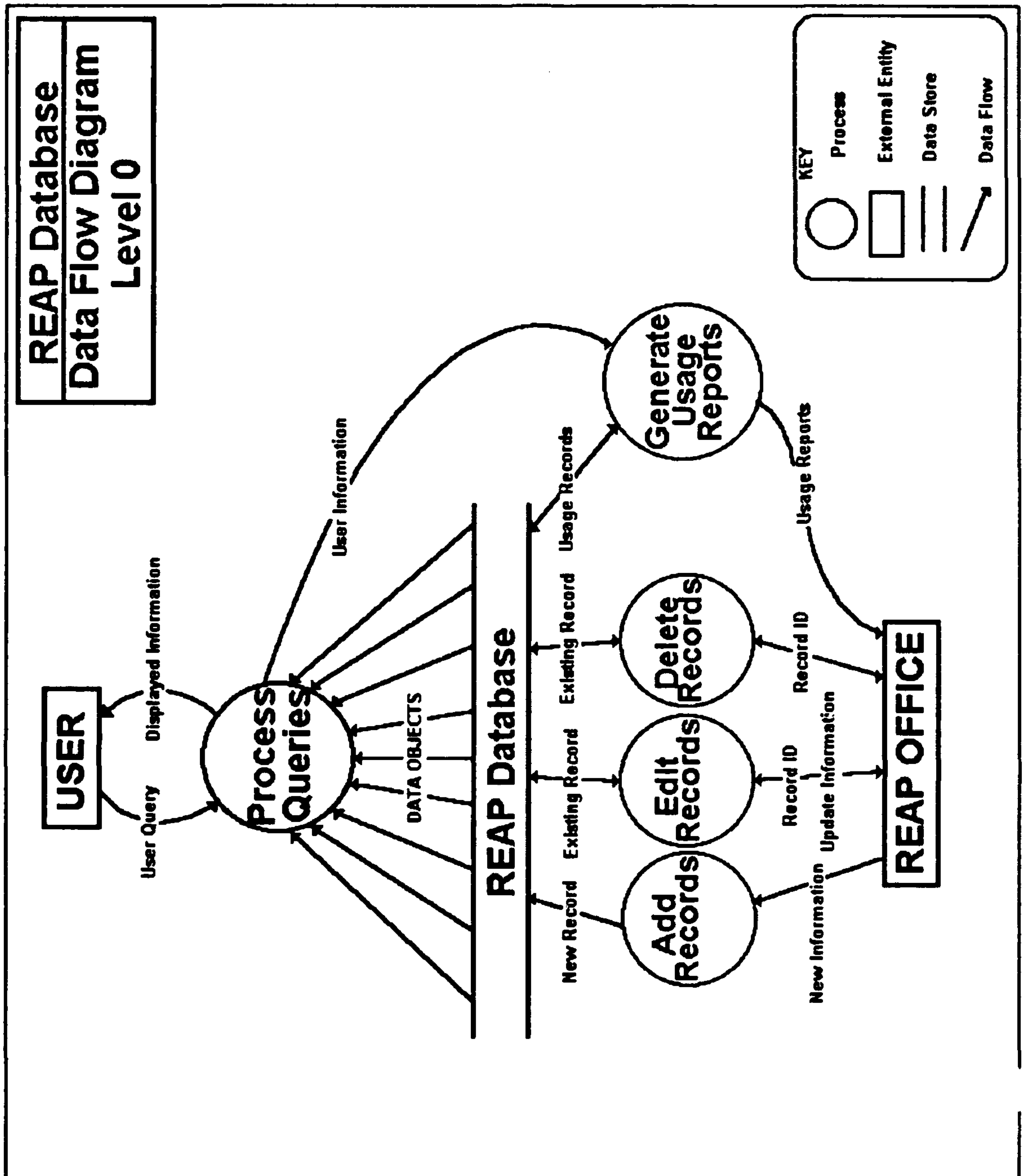

REAP Database
Data Flow Diagram
Level 0
USER
User Query
Displayed Information
Process Queries
User Information
DATA OBJECTS
REAP Database
New Record
Existing Record
Existing Record
Usage Records
Add Records
Edit Records
Delete Records
Generate Usage Reports
New Information
Record ID
Update Information
Record ID
Usage Reports
REAP OFFICE
KEY
Process
External Entity
Data Store
Data Flow

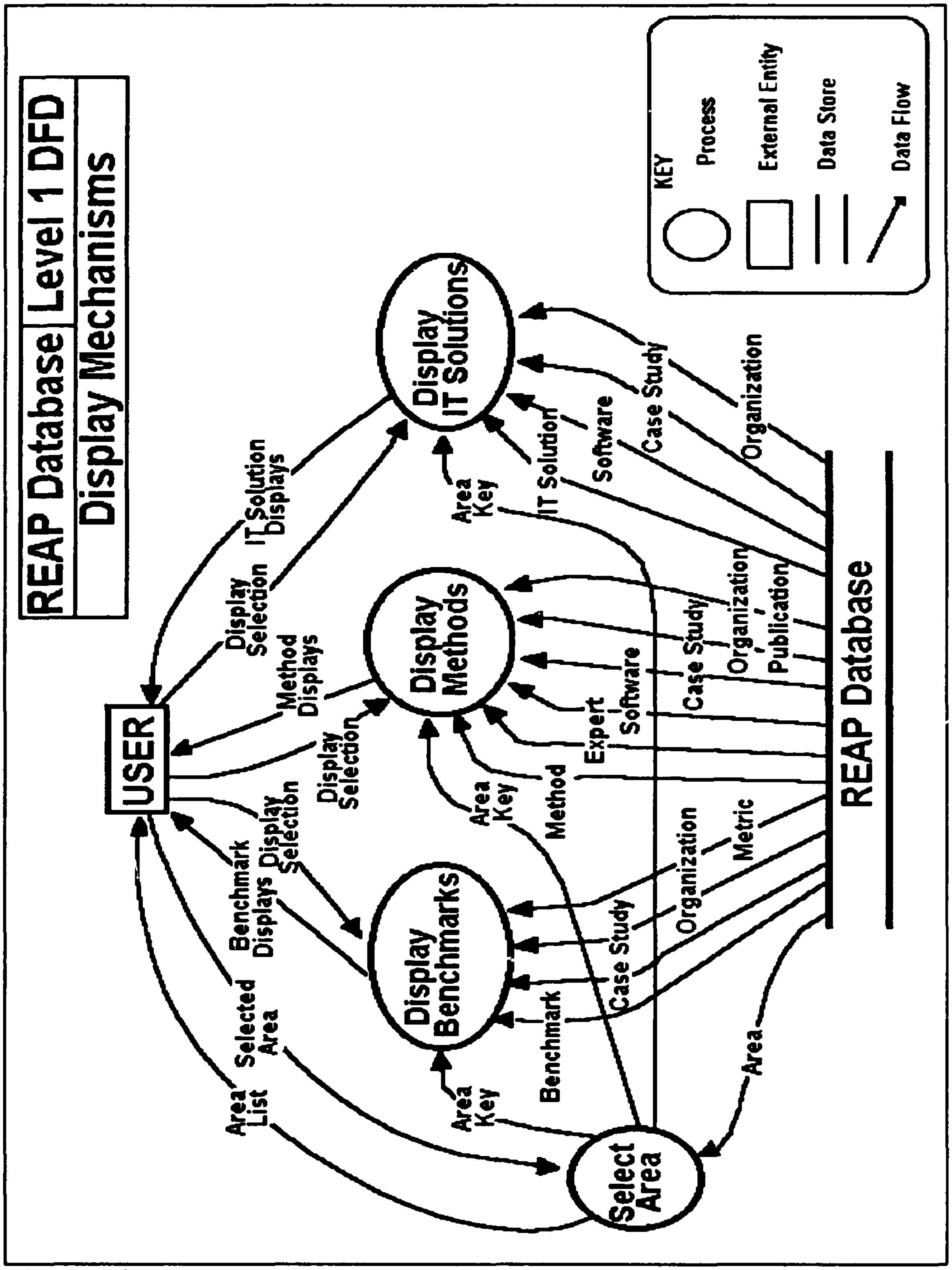
Figure 5

This task entails decomposing and defining the subprocesses contained within the Process Queries process.

The idea for the initial functional segregation of the Process Queries process comes from an analysis of the REAP database data objects. In general, there are three primary data objects, the Method object, the Benchmark object and the IT Solution object, which are vastly different in structure, although they share some common object properties. Instances of these data objects are linked by the Area object. It logically follows that a different mechanism is required to display information from each of the primary objects. Figure 6 illustrates the decomposition of Process Queries derived from this analysis approach. Initially, a mechanism is required to solicit and obtain the desired functional area information from the user and use this data to filter unrelated records from the information presented to the user. The Select Area process performs this function. Determining how the Select Area performs this task is important because, as will be seen, similar choosing and filtering functions are required throughout the REAP database application.

Since from a data-oriented stand-point, all the data objects in the database are contained either directly or indirectly in the Area object, when a user selects an Area instance, he/she is essentially deciding to view a summary of that Area instance. The focus of the analysis of the Select Area process is, "How a particular instance of the Area object is to be chosen for viewing?" There are two solutions to this question:

1. The user could view complete Area records (including all related object properties) sequentially.



REAP Database
Level 1 DFD
Display Mechanisms
USER
Display IT Solutions
Display Methods
Display Benchmarks
Select Area
REAP Database
IT Solution Displays
Display Selection
Method Displays
Display Selection
Benchmark Displays
Display Selection
Selected Area
Area List
Area Key
Area Key
Area Key
IT Solution
Software
Case Study
Organization
Organization
Publication
Case Study
Organization
Software
Case Study
Expert
Software
Method
Metric
Organization
Organization
Case Study
Case Study
Benchmark
Benchmark
Area
KEY
Process
External Entity
Data Store
Data Flow

Figure 6

2. The user could be presented with a list of only the Name property of all the Area instances and choose the record to be viewed.

While the first solution has the advantage of simplicity, it would most certainly prove too time consuming for the user. It is concluded that the second alternative is the better solution in terms of ease of operation for the user while not being significantly more complicated than the first solution. The Select Area process is thus defined as a process that compiles and presents a list of the names of Area instances and then allows the user to choose a particular instance for further review.

Display Methods, Display Benchmarks, and Display IT Solutions are identified as the subprocesses to carry out the tasks of displaying the primary data objects. Each of the "Display" processes will be made up of subprocesses in order to query and display instances of the their respective object properties. The decomposition of each display processes is the next step in the functional requirements analysis.

1. Decomposition of Display Methods

The Method object is the most complicated of all the data objects, containing four multi-valued object properties. Functional analysis of the Display Methods process must first address how a particular instance of the Method object is to be chosen for viewing and then how an instance of the Method should be displayed. When deciding how a Method instance should be displayed, it is necessary to consider how these object properties should be displayed and, perhaps more importantly, what mechanism is needed for the user to choose which instance of a multi-valued property she/he wants to see. As will be seen, these considerations are common to the analysis of all the display processes.

Thus, solutions formulated for the Display Methods process will carry over to the Display Benchmarks and Display IT Solutions processes.

The task of determining which Method instance to be chosen is similar to the task of determining which Area instance is to be chosen. A process similar to the Select Area process, where a only a list of the key fields of the Method object instances related to the chosen Area instance, is displayed. The Method instance to be viewed is selected from this list.

Once a specific Method instance has been chosen, a process is needed to present its properties as a summary of the method to the user. The process for displaying the Method instance summary has three basic tasks:

1. Display the non-object properties of the Method object instance.
2. Enable the user to choose a particular object property for viewing.
3. Enable the user to select a specific instance of a multi-valued object property for viewing.

In other words, the Method object display process first displays a summary of the Method instance by presenting non-object properties. User options at this point include viewing lists of the key fields from related Expert, Organization, Case Study and Publication object instances. The user can then pick one of these other object instances for review.

Finally, processes are needed to display the summaries of the various object properties contained within the Method object. The process for displaying the Organization object summary is fairly simple, requiring the presentation of non-object properties only. Because the Organization object is an object property of the Expert and

Case Study objects, the processes for displaying those two objects will both need a link to the Organization object display process. Like the Method object, the Publication object contains multi-valued object properties (Expert and Method). The Publication object summary display process must not only display the non-object properties of the Publication object, but list the key fields of related object property instances, allowing the user to choose one for review. It should be noted that since another Method instance can be chosen at this point, the search may circle back on itself. This iterative feature may be present problems during operation so safeguards (perhaps limiting the number of iterations) may need to be built into the process. The purpose of the Expert object property contained in Publication is to provide "About the Author" information. It is a multi-valued property because it is possible to have more than one author. However, it is unlikely that more than two or three authors will be associated with a specific publication. Therefore, it is concluded that in this situation, for the sake of functional simplicity, the user will have to view the summaries sequentially. Figure 7 depicts the data flow inside the Display Methods process. The data flow area Key comes from the Select Area process. The Select Method process uses this data flow to filter unrelated records when retrieving Method Object instances from the data store. The data flow Method List represents the key fields of these instances, presented to the user in a list format. The data flow Method Choice from the user specifies which instance is to be viewed. The key field of the Method instance that the User selects is passed to the Display Method Summary process in the Method Key data flow.

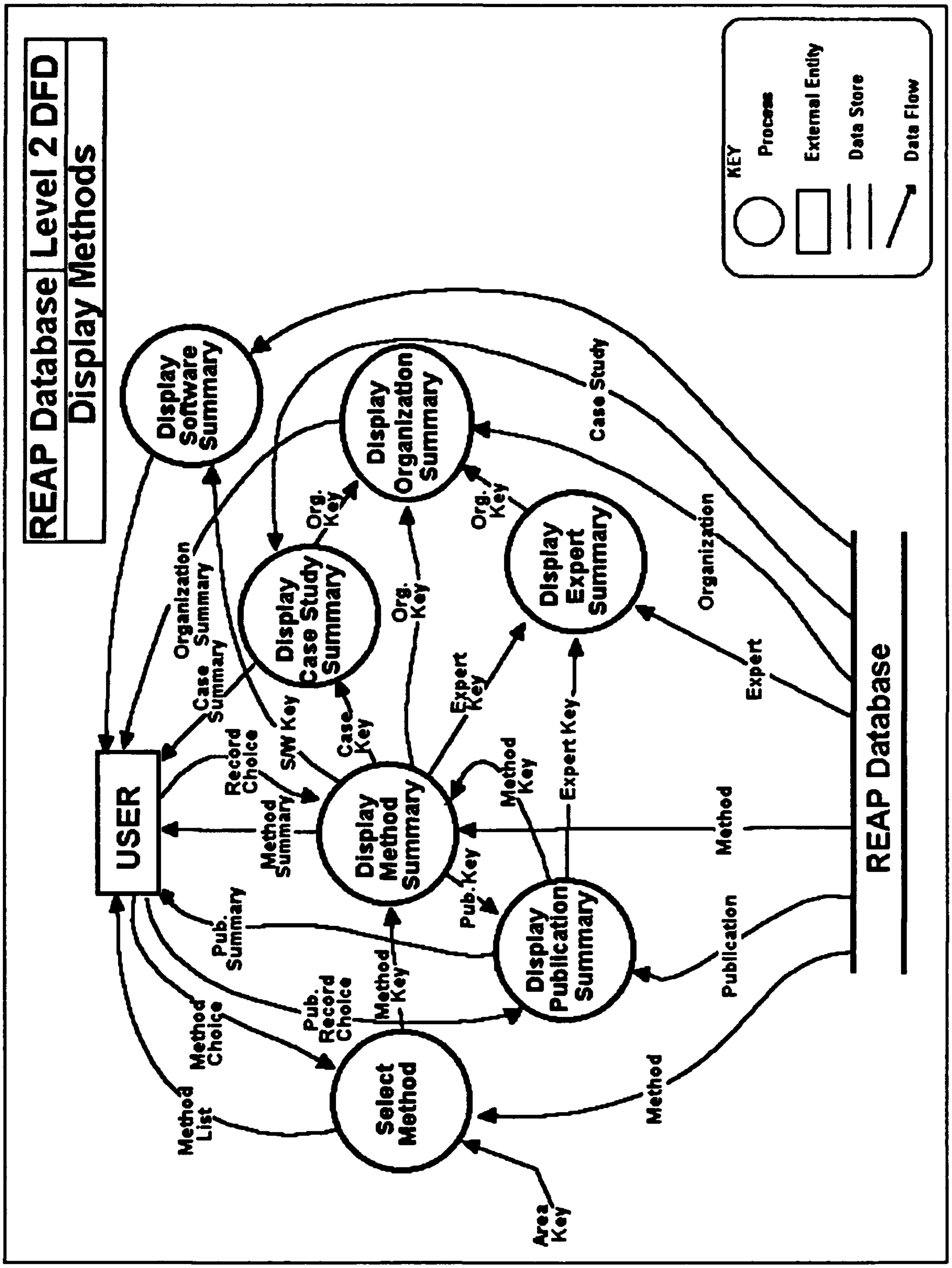

REAP Database Level 2 DFD
Display Methods
USER
Select Method
Display Method Summary
Display Software Summary
Display Case Study Summary
Display Organization Summary
Display Expert Summary
Display Publication Summary
REAP Database
Area Key
Method List
Method Choice
Method
Method Key
Pub. Record Choice
Pub. Summary
Method Summary
Record Choice
S/W Key
Case Summary
Organization Summary
Case Key
Org. Key
Expert Key
Pub. Key
Publication
Expert
Organization
Case Study
KEY
Process
External Entity
Data Store
Data Flow

Figure 7

The data flow Method Summary is all non-object properties and the key fields of the object properties related to a specific Method instance. Essentially, this is the non-object properties presented in a logical format and a list of object property instances for each of the four multi-valued object property (Expert, Organization, Case Study, Publication). It should be noted that the data flow diagram represents the logical flow of data in the system vice the temporal flow of data, i.e. all these properties of the Method object are not displayed to the user at the same time. The data flow Record Choice specifies which object instance summary is to be viewed. The data flow Pub. Record Choice serves a similar function for the Display Publication Summary process. The data flows between Display Method Summary and the other summary display processes represent the key fields, enabling the summary display processes to retrieve the correct object instance from the data store.

2. Decomposition of Display Benchmarks

Since the Benchmark data object contains only one multi-valued object property and two single-valued object properties, the Display Benchmarks process is less complicated than the Display Methods process. As with Display Methods, a sub-process is needed to present the users with a list of the applicable Benchmark object instances and allow them to select a record for viewing. A second process is needed to display the Benchmark object instance chosen. This process displays the non-object properties of the Benchmark object, displays single-valued object properties (Case Study and Organization) and enables the user to choose which Metric object instance to review. Sub-processes to

display summaries of the Case Study object, the Organization object, and the Metric object round out the Display Benchmarks process.

Figure 8 depicts the data flow in the Display Benchmarks process. As before the Area Key data flow provides the information on which the Select Benchmark object filters un-applicable records from the Benchmark List. Benchmark Choice from the user provides the user's viewing choice and is translated into the Benchmark Key data flow which is used by Display Benchmark Summary to retrieve the proper Benchmark record from the data store. The Display Benchmark Summary process provides all the non-object properties and the applicable key fields of the multi-valued object property Metric to the user in the data flow Benchmark Summary. It also passes key fields for the Organization and Case Study object properties to their respective display processes.

Key properties for the single-valued object properties are sent to the proper summary display process. The data flow Metric choice is used to select the Metric object record to be viewed.

3. Decomposition of Display IT Solutions

The IT Solution data object is fairly simple, containing only two multi-valued object properties, indicating that, like Display Benchmarks, the Display IT Solutions process will be less complicated than Display Methods. Like the previous display processes analyzed, Display IT Solutions needs a sub-process that allows the user to pick a specific record to review from a list of IT Solution instances related to a previously chosen area. A process that displays all the non-object properties of the chosen IT Solution instance along with the key fields of the object property instances is also needed.

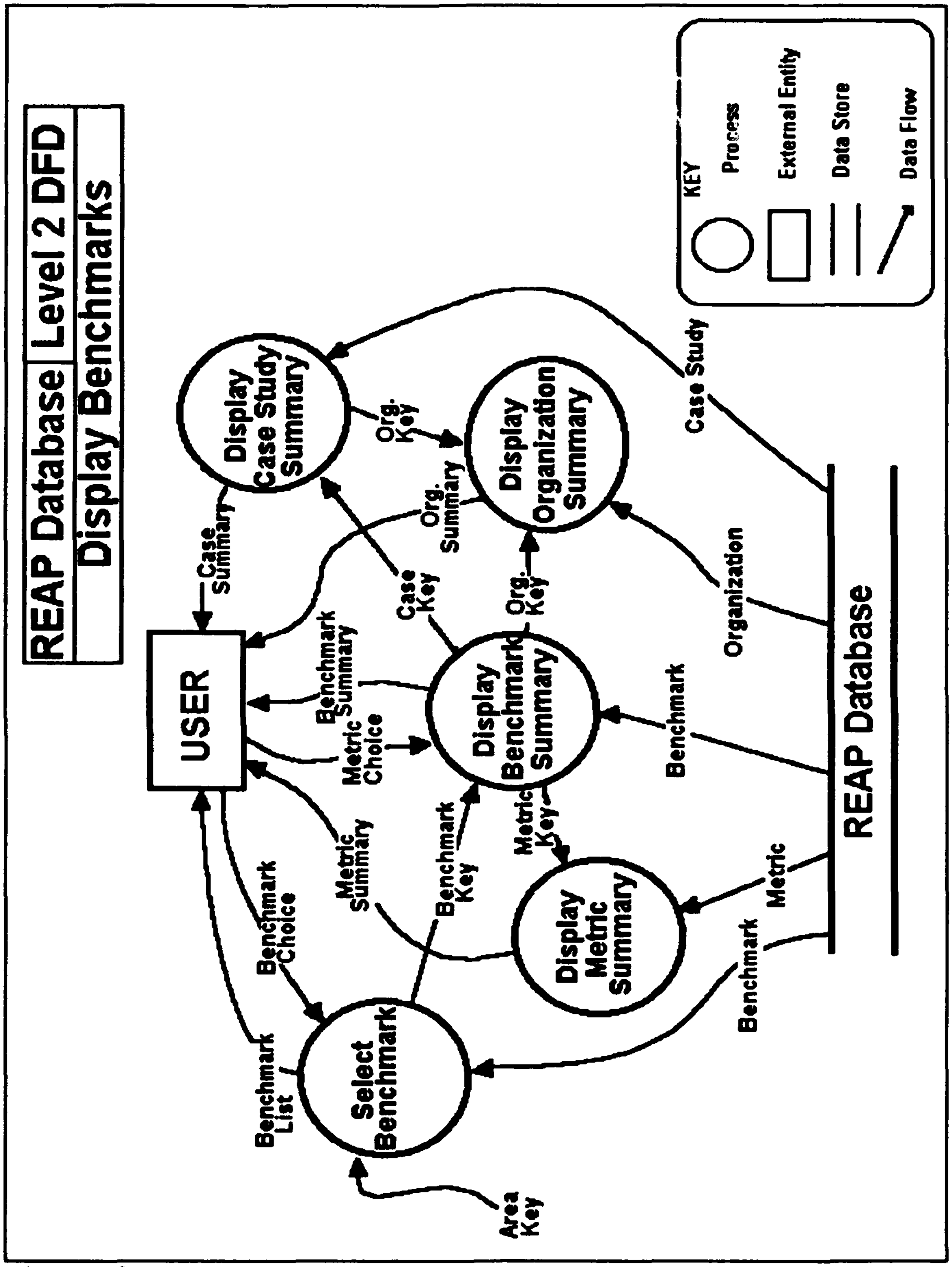

REAP Database | Level 2 DFD
Display Benchmarks
USER
Display Case Study Summary
Display Organization Summary
Display Benchmark Summary
Display Metric Summary
Select Benchmark
REAP Database
Case Summary
Org. Key
Org. Summary
Case Key
Org. Key
Benchmark Summary
Metric Choice
Metric Summary
Benchmark Key
Metric Key
Benchmark Choice
Benchmark List
Area Key
Case Study
Organization
Benchmark
Metric
Benchmark
KEY
Process
External Entity
Data Store
Data Flow

Figure 8

Finally, processes to display the related object property summaries are required.

Figure 9 defines the data flow within the Display IT Solutions process. As before, the data flow Area Key is used to set the filter for applicable IT Solution instances, the key fields of which are presented to the user (the IT Solution List data flow). The user's choice (IT Solution Choice) is translated into the key field of the selected record (IT Solution Key) and passed to the Display IT Solution Summary process for retrieval of the full record. The user's choice for viewing summaries of specific multi-valued object property instances is contained in the Record Choice data flow. This is translated into the correct key by the Display IT Solutions process and sent to the proper display process. It is interesting to note that the Display Organization process has no direct link to the Display IT Solutions process. Its key is taken from the specific Software or Case Study instance being viewed.

C. SPECIFICATIONS OF THE DISPLAY MECHANISMS

The last step in specifying the functional requirements is to define the display mechanisms in a formal manner. Kroenke and Dolan, [1988] break the specification of the display mechanisms into five areas:

1. A description of the output, identifying all the display screens needed.
2. The identification of the source data for the display.
3. Notes on the processing needed in order to produce the displays.
4. An estimate on the volume of usage. The number of times the display will be used each time the application is run.

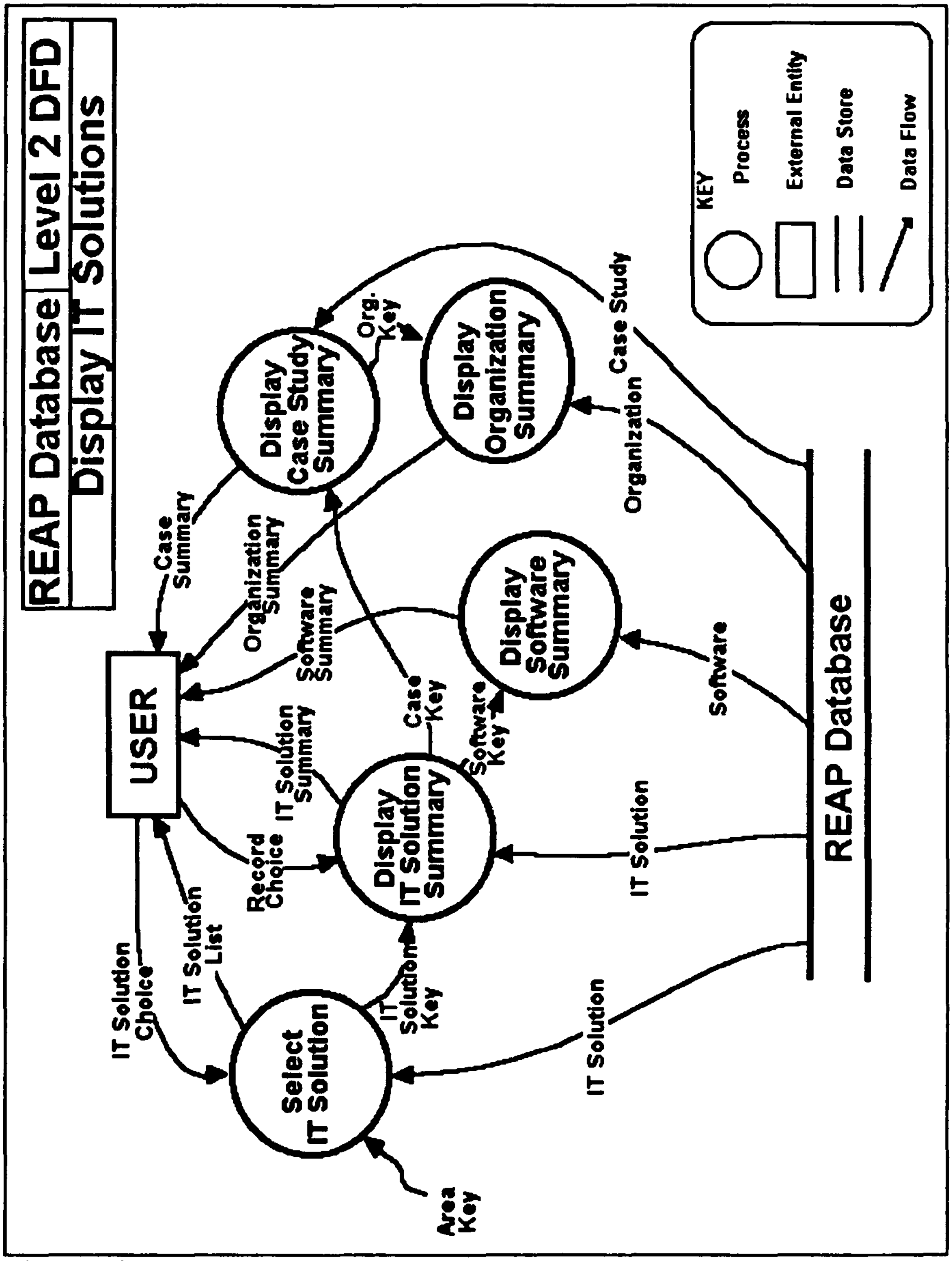

REAP Database
Level 2 DFD
Display IT Solutions
USER
Select IT Solution
Display IT Solution Summary
Display Software Summary
Display Case Study Summary
Display Organization Summary
REAP Database
Case Summary
Organization Summary
Software Summary
IT Solution Summary
Record Choice
IT Solution List
IT Solution Choice
IT Solution Key
Case Key
Software Key
Org. Key
Area Key
IT Solution
IT Solution
Software
Organization
Case Study
KEY
Process
External Entity
Data Store
Data Flow

Figure 9

5. An estimate of the frequency of use. How often the display mechanism is likely to be used.

Functional specifications for the mechanisms shown in the level 1 data flow diagram (Figure 6) provides a comprehensive description of the functional requirements. These specifications are sufficient to describe how the user controls the database queries, and how queries and displays are coordinated to provide the user with the right information at the right points in the search. The functional specifications are listed in Appendix B.

VII. EVALUATION PHASE

A. EVALUATION PHASE OVERVIEW

According Kroenke and Dolan [1988], the evaluation phase consists of three tasks:

1. Reassess the feasibility of the application in light of the requirements specifications.
2. Identify alternative application system architecture.
3. Reevaluate user requirements within the context of each possible solution.

The solution that best meets the needs of the requirements is chosen as the architecture for the design of the database.

B. REASSESSMENT OF REAP DATABASE FEASIBILITY

The three feasibility issues addressed in the definition phase which affect the development of the REAP database prototype, software availability, hardware availability and time constraints, are still relevant concerns. The basic factor determining the prototype feasibility with respect to these areas is database size. The initial assessment was that the prototype database would consist of 50-80 records of varying sizes. A reassessment of these figures, based on the requirements specifications is needed.

Another feasibility issue not addressed during the original, definition phase assessment is the availability of data for inclusion into the database. In order to address this issue a literature search for publications, case studies, experts and organizations

dealing with business process redesign was conducted. In addition to answering the data availability question, information gathered in the literature search had a direct impact on the issue of database size.

1. Results of REAP Literature Search

The general purpose of the search was to locate and compile information on business re-design methods and best business practices. Specifically, the search was to find:

1. Business redesign methods (descriptions and case studies).
2. Business process benchmarks and metrics.

The search was conducted over a period of twelve weeks. The following sources were canvased:

- Naval Postgraduate School, Dudley Knox library
- University of California library system's MELVYL computer catalog
- Computer Select database
- Computer Aided Manufacturing International (CAM-I), Cost Management System (CMS) program publications
- Contacts with business improvement consulting firms
- NPS sponsored seminars
- USENET news groups (unofficial forum provided via Internet/DDN)

Seventy-six literature items were found that could provide information for the REAP database. The bulk of the results of the literature search were articles on business process improvement and performance measurement, found in a variety of periodicals.

a. Business redesign methods

The research indicated that, while business re-design/reengineering is accepted as very effective means to improve the efficiency and effectiveness of an organization, most companies that conduct business reengineering develop their own, in-house, sometimes *ad-hoc* plan. Formal business reengineering methods remain largely undeveloped. Only two formal re-design/reengineering methods were discovered. One is the ISP system developed by the U.S. Army Corps of Engineers. As explained previously, it is a structured process for conducting process reengineering using IDEF tools and prototyping methods. A second, less dramatic, in-house system is called "Painting the Bridge", developed by the USAA insurance company. In Painting the Bridge, a team of organizational experts "starts at one end of the company and goes through it one division at a time, with an eye towards organizational health and organizational development...doing away with unnecessary work, titles and fiefdoms" [Teal,1991]. The team completes the cycle every two years and then begins at the start again, similar to the manner in which bridges are painted (from one end to the other then back to the start).

While not strictly re-design/reengineering methods, four relatively new business process evaluation/management methods were researched that play important roles in business process re-design:

1. *Total Quality Management (TQM)* - a management method "aimed at providing the highest levels of quality, productivity, flexibility, responsiveness, and customer satisfaction. It forms a participative management style [and] networks all of the

people and processes in harmony with each other and the [business] environment. It ensures a sound system of analysis to cope with the many changes that a business will see[.]" [Shores, 1990]

2. *Activity Based Costing (ABC)* - a way of accounting aimed at identifying all the costs related to a specific product and determining why and how they are incurred.

> "ABC reveals the links between performing particular activities and the demands those activities make on an organization's resources...ABC analysis helps managers focus their attention and energy on improving activities that will have the biggest impact on the bottom line." [Cooper and Kaplan, 1991]

3. *Benchmarking* - "the continuous process of measuring products, services, and practices against the toughest competitors or those companies recognized as industry leaders (David T. Kearns, CEO, Xerox Corporation)" [Camp, 1989].

4. *Business Process Improvement (BPI)* - "a systematic methodology developed to help an organization make significant advances in the way its business processes operate." BPI focuses on eliminating waste and bureaucracy and provides a system that will aid in simplifying and streamlining operations while ensuring good output. [Harrington,1991]

b. Business benchmarks and metrics

Finding business benchmark organizations and information on their benchmark processes proved unexpectedly time consuming and difficult. It is concluded that benchmarks should only be included in the REAP database if dollars are allocated to identify benchmark organizations. Alternatively, at least one organization was discovered (the American Productivity and Quality center; APQC) that provides a

benchmarking clearing house database and referral service. This service is proprietary, but may be cost effective. An examination of benchmark researching costs seems warranted.

The search found fewer case studies that dealt with military or DOD organizations than case studies of civilian organizations. Given that this situation is probably true as a whole and given that DOD managers will probably find it easier to associate with military case studies than civilian ones, it will be important for the maintainers of the deployed REAP database to be especially vigilant in searching for military case studies. One idea may be to require reports from DOD units undergoing process redesign/reengineering and use the best examples as new case studies.

c. *Conclusions drawn from the literature search*

From the standpoint of data availability, the REAP database appears feasible. Most of the desired process redesign information (case studies, benchmarks, general methods, and metrics) can be quantified in such a way as to be useful in a database format. More detailed information can be summarized and combined with information on how to obtain source publications for inclusion in the database. While there are no technical reasons that benchmarks cannot be included, it may be cost effective to obtain benchmarking services from an outside source.

2. REAP database size

In determining the effect that database size will have on the feasibility, two issues must be addressed. First, the likely size of the prototype database will effect the

feasibility of adhering to the completion date requirements. Second, the size of the full scale database may affect the choice of deployment hardware and software. The results of the literature search were used to establish the number of records to be included in the prototype database. Additionally, during the literature search, an idea of the amount of business process redesign information available as a whole was developed leading to the decisions regarding the eventual size of the database. In order to get a complete idea of the database size, a small amount of design work was necessary in order to establish the number and types of data files necessary. While a more detailed explanation is provided in the design phase, it should be noted that data files necessary to establish the data object relations specified in the requirements were developed. Table 4 provides detailed estimates of the size of the prototype and full scale databases.

Table 4.

REAP DATABASE SIZE ESTIMATE
(Number of Records)

Key: D -Desired E - Estimated R - Required

Data File	Proto Size	Full Size	Reasons
Area	18	18	CIM Functional Areas
Method	8	15	Literature Search Results(Projected)[1]
IT Solution	15	15	Literature Search Results
Benchmark	6	72	4 per Area record D[2]
Case Study	18	147	1 per Benchmark R, 2 per IT Solution D, 3 per Method D
Expert	85	358	2 per Method D, 1.22 per Publication E[3] minus Authors w/ multiple Publications (8%) E[4]
Organization	28	371	1 per Case Study R, 3 per Method D, .5 per Expert E
Software	20	165	10 per IT Solution D, 1 per Method E
Metric	2	18	.25 per Benchmark E[5]
Publication	51	288	20 per Method D/E[6] minus Publications covering 2 or more methods (8%) E[7]
Data Files Needed to Establish Relations			
Area-Method	72	72	4 Applicable Methods per Area D
Area-IT Solution	90	90	5 IT Solutions per Area D
Benchmark-Metric	2	108	1.5 Metrics per Benchmark E[8]
IT Solution-Case Study	6	30	2 Case Studies per IT Solution D
IT-Solution-Software	15	23	1.5 Software per IT Solution D
Method-Case Study	11	45	3 Case Study per Method D

Table 4.

REAP DATABASE SIZE ESTIMATE
(Number of Records)

Key: D -Desired E - Estimated R - Required

Data File	Proto Size	Full Size	Reasons
Method-Expert	0	30	2 Expert per Method D
Method-Organization	6	45	3 Organization per Method D
Method-Publication	70	300	20 Publications per Method E[6]
Method-Software	5	8	.5 Software per Method D
Publication-Expert	85	351	1.22 Expert per Publication E[3]
Total	613	2569	

Notes:

1 Literature search; 8 Methods discovered. Projected at least 50% more undiscovered.
2 2 DOD benchmarks; 1 Government benchmark; 1 Industry Benchmark.
3 Literature search results; 22% of Publications had two or three authors.
4 Literature search results; 8% of authors had authored another publication included in the search.
5 General estimate; Common metrics are used to measure multiple business processes.
6 Literature search results; The number of publications covering a specific method ranged from 1 to 18. The mean of the 7 data points was 10. Double this is considered an adequate goal.
7 Literature search results; 8% of the publications included covered 2 or more methods.
8 General estimate; Every other benchmark can be measured by two or more metrics

In order to estimate the database sizes in bytes, sample data files were developed on a personal computer (IBM-compatible/MS-DOS system). Based on the requirements specifications, three type of files were identified. The first type is a file containing a few (2-3) simple fields and a long text field (e.g. Method), the second type

is a file containing more simple fields (3-5) and longer or more text fields (e.g. Publication), and a the third type is a file containing a few (2-5) simple fields and no long text fields (e.g. Metric or Publication-Expert). A number of test records were developed and the size of the resulting files measured. These sizes were used as estimates for computing the eventual size of the prototype and full scale REAP databases. Table 5 provides these size estimates.

Table 5.

REAP DATABASE SIZE ESTIMATE
(Size in bytes)

Type of file	Prototype Size	Full Size
Type 1	112,048	286,080
Type 2	243,468	1,638,160
Type 3	41,630	126,656
Total	397,146	2,050,897

Even allowing one to two megabytes of storage space for the REAP database application and five to ten megabytes for the database management system software (a conservative estimate), it is would be physically feasible to develop the prototype and full REAP applications using systems from late model personal computers up to mainframe computers. Most commercial relational database management systems are capable of handling a system of these sizes. The number of records considered for inclusion in the prototype based on the requirements and the literature search (613), is many more than the original assessment (80). However, over half of these records will belong to simple files and will not consist of more than two to three short fields. As for the rest, since the

purpose of the prototype is validate the data structure and develop the user interface, the number of records actually entered could be reduced in order to produce the prototype by the desired completion date. Therefore, it is concluded that, despite the great increase in the potential size of the prototype database, it is still feasible to produce the prototype in the time allotted. Additionally, the potential size of the full scale REAP database does not effect its feasibility.

C. ALTERNATIVE APPLICATION ARCHITECTURE

In order to comply with the "on-line" query capability specified in the application requirements, there are two practical REAP database application architectures: a stand alone PC-based system and a mainframe based system providing access to users via the MILNET (MILitary NETwork; the unclassified part of the Defense Data Network - DDN - that is connected to the Internet). It is understood that an office is to be opened under the DDI which will be responsible for administering the REAP database and conducting continuing business process improvement research aimed at providing periodic updates to the database. This organization is called the REAP office. How the user gets this information from the REAP office is dependent on the system architecture chosen. Three questions need to be addressed when specifying the system architecture:

1. How will the user access the database?
2. How will the records in the "user's" database be maintained and updated?
3. What type of hardware and software will be needed to support the database?

1. Architecture for a PC-based application

Personal computers (PCs) are in use virtually everywhere in DOD. Deploying the REAP database as a PC based system means that users will access the REAP database by running copies of the REAP application on their own PCs. Providing these multiple copies should not pose any significant technical or administrative problems for the REAP office. The challenge for the REAP office will be to maintain (fix application errors and provide data updates) the multiple copies deployed throughout DOD.

For all practical purposes, there are only two types of PCs: IBM compatible machines (Intel 80X86 CPU running DOS or OS-2) and Macintosh computers (Motorola CPU running Apple System 6.x or 7.0). A significant factor to be addressed when considering a PC based architecture is that, because of differences between these two types of computers, two REAP applications would have to be developed and maintained. While there is an application (Soft PC) which will allow DOS based applications to run on Macintosh machines, it is felt that most Macintosh users would find it simpler and cheaper to run a Macintosh based application. To avoid the risk of losing these potential users, a Macintosh based application is recommended as well as a DOS application.

In determining the means of transmitting the REAP database application and data to the users, two solutions are possible: First, should a user have access to the MILNET, the appropriate files could be electronically transmitted from the REAP office using the File Transfer Protocol (FTP) function. Likewise, the correspondence necessary to establish the subscription to the REAP database could be done via e-mail. A second

alternative for PCs not linked to the MILNET would be to send the files on floppy disks via the U.S. postal system. Subscription administration could be conducted via postal mail or telephone. It should be noted that these two possible solutions are not mutually exclusive. The electronic dissemination would be the preferred method (cheaper and faster for the REAP office) and the floppy disk method would be used should the user's PC not have MILNET connections.

Hardware requirements for this architecture are a hard disk, and enough memory to run the database management system (DBMS) used to implement the REAP database. Central Processing Unit (CPU) speed should not be an issue. For the simple types of queries that the REAP database performs and for the number of records involved, even older, slower (8-10 Mhz clock speed) PC should provide acceptable search times. The hardware requirements should not exclude many PC platforms. The only software requirement would be a legal copy (license to run the software) of the DBMS. There are currently a number of commercial DBMS packages available ranging in price from several hundred dollars to over a thousand dollars. Examples of PC based DBMS's include dBase IV (IBM-PC), Foxbase (IBM-PC, Mac PC), Clipper (IBM-PC) and Paradox (IBM-PC).

2. Mainframe Application Architecture

A mainframe based system providing access via MILNET is a much simpler solution from the REAP office administration standpoint. Only one REAP application would have to be maintained. Data updates could be performed on a daily basis thus ensuring the latest information for the user. Access to the REAP application would be

provided by a communications protocol called TELNET that is part of the network protocol (TCP/IP; Transmission Control Protocol/Internet Protocol) suite used by MILNET. Postel and Reynolds [1983], the authors of the TELNET specifications, describe the function of TELNET as providing "a standard method of interfacing terminal devices and terminal-oriented processes to each other." In other words, once a TELNET link is established to the mainframe hosting the REAP database, the user could run the database application from his/her own terminal.

Obviously a user would need access to a MILNET-connected host computer to obtain access to the REAP database. The availability of access for potential users of the REAP database will directly affect its success. According to the DDN Network Information Center (NIC) database [Network Information Center, Aug 1992] there are 1,034 host computers tied into the DDN worldwide. Additionally, there are 220 DDN Terminal Access Controllers (TACs) in the United States which provide local call-in accesses to the DDN for modem-equipped PCs. Should a potential user not have access to a host computer, it may be possible for the REAP office to arrange TAC access authorization.

For this architecture the only hardware requirements for the user are a computer or terminal that had MILNET access. There are no specific user software requirements. The hardware requirements for the REAP office are a mainframe computer with the proper communications connections to MILNET. The software requirements include installation of TCP/IP and a suitable DBMS. Examples of mainframe DBMS's include INGRESS and Oracle.

D. REEVALUATION OF REQUIREMENTS

The final step in the evaluation phase is to reevaluate the requirements specifications in the context of each solution alternative. The goal of this final step is to select the best alternative. However, it is Kroenke and Dolan's [1988] position that if all the requirements cannot be accommodated during the present project, then priorities should be set and some requirements deferred for future projects. The understanding is that the best solution should always be pursued, even if it is not feasible to develop the entire application at once.

The criteria by which the two possible solutions are measured are:

1. *Maintainability* - In the context of the solution, how easy or difficult will it be to maintain the database application and provide updates to the users.
2. *Functionality* - What features of a solution either enhance or obstruct a user's easy access to timely information.
3. *Cost* - What costs will likely be incurred, in most cases, in order to give a user access to the database and maintain that access.

1. Reevaluation of requirements in a PC-based system context

Three factors will affect the maintainability of the REAP database if deployed as a PC-based system. First, the REAP office will have to maintain two REAP applications: one for IBM compatibles and one for Macintosh machines. This will probably mean twice as much work for the REAP office. Second, the REAP office will probably be called upon by the users to troubleshoot problems caused by unforeseen hardware and/or operating system incompatibilities with the DBMS, the REAP

application or both. In effect, the REAP office will not just be maintaining two applications, but send each copy of the REAP application to the users. Third, it will probably be economically feasible only to send fixes to common minor problems at periodic intervals vice sending them as soon as they have been developed. This will mean that users will have to "work around" problems with the application until the next periodic upgrade is published.

The user interface capabilities of a PC-based system would certainly enhance the ease of access to REAP information. Many PC-based DBMS applications offer a variety of user interface features including multi-color displays, pop-up menus and help functions, and even graphical user interfaces (GUI). These features can be combined to produce an effective user friendly interface for the REAP application. The major functionality weakness of a PC-based system is that the information contained in the database will be outdated from the time the user receives it. Timeliness of the data will depend on how often the REAP office to makes periodic updates and will vary from user to user. It will be impossible to predict how outdated REAP information can be and still be considered useful, until the database has been deployed and user statistics can be gathered. However, the conclusion can be drawn that, should timeliness of information be found to be an important factor in determining database usefulness, a PC-based system will be deficient in this area.

For the user, the principle costs of a PC-based architecture include the acquisition of a copy of the DBMS to run the REAP application. It is assumed that the user will already have the PC on which to run the database. For the REAP office, costs

will include those associated with maintaining two software applications running on different platforms as well as the overhead costs involved in maintaining a subscription system (bookkeeping, floppy disks, postage, etc.)

There is nothing in the PC-based architecture that would preclude the full development of a REAP database prototype in the time allotted.

2. Reevaluation of Requirements in a Mainframe-MILNET Context

Under a mainframe-MILNET access architecture, REAP application maintenance is vastly simplified. The REAP office will be responsible for maintaining one copy of a single REAP application, namely the copy running on the mainframe. Descriptions of problems encountered by the users can be e-mailed to the REAP office. Solutions can be implemented as soon as they have been developed and tested. The limits of the TELNET protocol do have a severe impact the functionality of the REAP user interface. In order to operate between as many different computer systems as possible, TELNET provides a single format, called Network Virtual Terminal (NVT), that all systems must use to communicate with each other during TELNET sessions. Postel and Reynolds [1983] state:

> "An NVT is an imaginary device which provides a standard, network-wide, intermediate representation of a canonical terminal. This eliminates the need for "server" and "user" hosts to keep information about the characteristics of each other's terminals and terminal handling conventions. Both user and server, map their local device characteristics and conventions so as to appear to be dealing with an NVT over the network."

Figure 10 provides a graphical representation of the NVT concept.

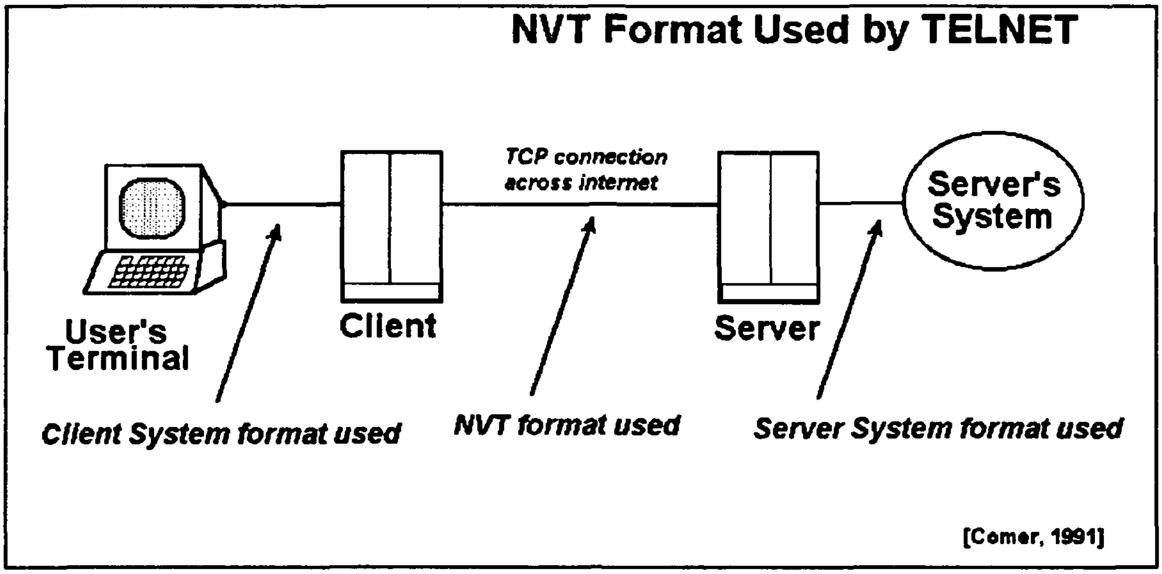


Figure 10

The NVT protocol poses severe restrictions on the type of format that the REAP database user interface mechanisms can employ. First, the NVT code set is seven-bit US ASCII in an eight-bit field. This effectively eliminates the possibility of employing a graphical user interfaces (GUI), which would need a more sophisticated, binary code set. Display screens are limited to ASCII text formats. Second, the NVT is essentially as a half-duplex device operating in a line-buffered mode [Postel and Reynolds, 1983], meaning that data is passed across the network one line at a time. This eliminates the possibility of using full screen ASCII based interfaces. Some type menu and command interface is the only remaining possibility. Although this type of user interface cannot be considered sophisticated, it can be design to be fairly "user friendly" and is adequate for the task. Examples of successful database interface applications that operate under the TELNET/NVT protocol include the NIC DDN Information database

and MELVYL, the University of California's library catalog system. A positive feature of the mainframe/MILNET access architecture is that data updates can be made whenever needed. This will mean that the user can always be provided with the latest REAP information.

Assuming that the user already has access to MILNET, there will be no additional costs, other than on-line time charges, to access the REAP database. There may be a fee involved in obtaining TAC access should a user require it. Costs to the REAP office will include a license fee for the DBMS and probably a usage fee for access to a mainframe host computer. It is assumed that the DDI's office already has a DBMS license and access to a mainframe and that these agreements need only be expanded to include the REAP database.

The functionality of a mainframe-MILNET based REAP prototype will be limited to implementation of the data structure. Both the prototype and full scale system are to be implemented in Oracle. The user interface features developed will be sufficient to demonstrate the relations in the data structure but will not be representative of what is desired for the full scale database. Implementation of user interface requirements should be deferred for a later project.

E. CONCLUSIONS

Despite some functionality drawbacks, the mainframe-MILNET access solution is the best architecture for implementation of the REAP database. The maintenance requirements of the mainframe-MILNET solution are far simpler than those for the PC-

base solution. Additionally, under the PC-based solution, the user bears more of the costs of acquiring the REAP database (DBMS license and possible subscription fee) and may therefore be more inclined not to acquire it. This is an important factor to consider because the success of the REAP database is dependent on its wide acceptance and use throughout DOD. Finally, the mainframe-MILNET access solution will generally provide more timely information than can be expected from the PC-base solution. Taken together, it is felt that these factors sufficiently outweigh the drawbacks of the mainframe-MILNET solution, namely a less sophisticated user interface than what would be available on a PC. It is important to reiterate that, although this solution is best in the long run, most features of the user interface will be deferred for a later project. However, a design for a user interface that meets the requirements specifications will be developed in the design phase of the initial REAP prototype.

VIII. DATABASE DESIGN

Kroenke and Dolan [1988] call for a two-part design phase:

1. Development of the database design
2. Development of the application design

This chapter will focus on the first part, the development of the database design. The objective of the database design effort is to draft the blueprint for the database structure from which the physical database design can be developed [Kroenke and Dolan, 1988]. For the REAP database, the blueprint will consist of a data relation diagram and data relation definitions.

A. OVERVIEW OF THE DATABASE DESIGN METHODOLOGY

The REAP database will be developed as a relational database. Kroenke and Dolan [1988] provide a description of the relational database concept:

> "data is stored in two dimensional tables called relations. Each row in the table represents a record [or instance]. Each column represents a field. A row is called a tuple. A column is called an attribute."

Pressman [1992] further illustrates the concept stating that the attributes take one of three characteristics. They can be used to identify an a record, describe a record, or make reference to another record in another table. The REAP database relations will be developed from the data objects, specified during the requirements phase.

1. Data Relation Normalization

While the relational model is a powerful concept in database design, care must be taken to design the data relations correctly. Kroenke and Dolan [1988] describe the principle effects of common design weaknesses and errors as modification anomalies. A modification anomaly occurs when an attribute is inappropriately included in a relation. The result is that the relation's data cannot be modified (instances deleted, changed or added) without data being lost or uselessly duplicated. To eliminate these problems, a process called normalization must be conducted as the data objects specified in the requirements phase are developed into data relations. Pressman [1992] provides four rules to follow when conducting this process.

1. A given instance of a [relation] has one and only one value for each attribute.
2. Attributes represent elementary data items; they should contain no internal structure.
3. When more than one attribute is used to identify a data object, be sure that descriptive and referential attributes represent a "characteristic of the entire object and not a characteristic of something that would be identified by only part of the identifier" [Schlaer and Mellor, 1988]
4. All non-identifier attributes must represent some characteristic of the instance named by the identifier and describe some other attribute that is not an identifier.

The goal of following these rules is to design relations that are in domain key normal form. Simply stated, a relation is in domain key normal form if it contains no modification anomalies [Kroenke and Dolan, 1988]. While there is no formal method

for developing data objects into relations that are in domain key normal form [Kroenke and Dolan, 1988], adhering to Pressman's rules and remaining alert for signs of modification anomalies should result in REAP database relations free of modification anomalies.

2. Entity Relationship Diagram Overview

Pressman [1992] states, "The cornerstone notation for data modeling is the entity relationship diagram." The purpose of the entity relationship diagram is to represent data relations graphically. A simple format is used where a rectangle represents a data relation and special lines represent the relationship "connections" between relations. An examination of the data objects will reveal what type of relationship exists between relations, either a one-to-many, a one-to-one or a many-to-many. Entity relationship diagrams can only represent one-to-many (triangle at the base of the many side of the connection) relationships or one-to-one (a simple line) relationships. Many-to-many relationships cannot be directly represented as one-to-many and one-to-one relationships are, because to do so will result in modification anomalies [Kroenke and Dolan, 1988]. In order to accommodate the existence a many-to-many relationship between two relations, a third relation, called an intersection relation, is created which contains the key fields of two principle relations. Two one-to-many relationships are established between the principle relations and the intersection relation. Mandatory relationships, where the existence of a an instance in one relation is determined by the existence of a related instance in a second relation, are designated by a hash mark across the connection line between the two relations, closest to the second

relation (the relation that determines the instance existence). Optional relations, where no such determination exists, are designated by a circle on the connection line closest to the second relation. Once completed, the entity relationship diagram forms the basis for the relation definitions.

3. Defining Relation Definitions

Relation definitions define the columns (attributes) of a relation. They provide the a name for each attribute and describe its domain. The attribute that uniquely identifies a record is designated as the key attribute. Should more than one attribute be needed for this purpose, the result is a combination key. Additionally, attributes that are used to establish relationships with other relations are identified as foreign keys.

B. DEVELOPMENT OF AN ENTITY RELATIONSHIP DIAGRAM

Examination of the ten REAP data objects reveals that each contains at least one multi-value relationship with another data object. In all there are twenty-four multi-value object properties contained in other objects. A more detailed examination reveals that these twenty-four multi-value object properties break out into eleven man-to-many relationships and two one-to-many relationships. It is the many-to-many relationships that need to be simplified into one-to-many relationships. This means that eleven intersection relations are required. Table 6 summarizes the intersection relations needed.

Table 6.

REAP DATABASE
MANY-TO-MANY RELATIONSHIP RESOLUTION

Relation 1	Relation 2	Intersection	Attributes
Area	Method	Area-Method	Area-Name, Meth-Name
Area	IT Solution	Area-Solution	Area-Name, IT-Name
Benchmark	Metric	Bench-Metric	Bench-Name, Metric-Name, Value [1]
IT Solution	Case Study	IT Solution-Case	IT-Name, Case-name
IT Solution	Software	IT Solution-S/W	IT-Name, App-Name
Method	Expert	Method-Expert	Meth-Name, Last-name, First-Name, MI [2]
Method	Software	Method-S/W	Meth-Name, App-Name
Method	Organization	Method-Org	Meth-Name, Org-Name
Method	Publication	Method-Pub[3]	Meth-Name, Pub-Title
Method	Case Study	Method-Case	Meth-Name, Case-Name
Publication	Expert	Pub-Expert	Pub-Title, Last-Name, First-Name, MI

Notes:
1. The attribute Value is included to associate the value magnitude of benchmark process with the metric used to measure it and the benchmark itself.
2. Expert has a combined key; Last-Name, First-Name, MI.
3. This intersection relation will serve two purposes, relating Publication records to a the specific Method instance being viewed and relating Method instances to the specific Publication instance being viewed.

With the many-to-many relationships resolved, the focus turns to identifying the one-to .any and one-to-one relationships in the REAP database. In addition to the two

one-to-many relationships previously discovered, there are two one-to-one relationships among the data objects. Table 7 defines these relationships.

Table 7.
REAP DATABASE ONE-TO-MANY AND ONE-TO-ONE RELATIONSHIPS

Relation 1	Relation 2	Type
Area	Benchmark	One-to Many
Organization	Expert	One-to-Many
Benchmark	Organizatio n	One-to-One
Benchmark	Case Study	One-to-One

Based on Tables 6 and 7, an entity relationship diagram can be constructed (Figure 11). In all, there are twenty-one relations in the REAP database. A testament to the complexity of the database is the fact that over half of these relations are intersections. In all, twenty-six relationships are described. Nineteen are mandatory-optional, meaning that the existence of a first relation record is dependent on the existence of a second related relation record, but the relation of the second instance is not dependent on the first. This is seen when dealing with intersection relations where the existence of an intersection relation record is always dependent on the existence of related records in the two other relations but the existence of a record in one of the other relations is not dependent on the intersection record. For example, every Area-Method record must have a related Area record; but should there be an Area record for which there are no

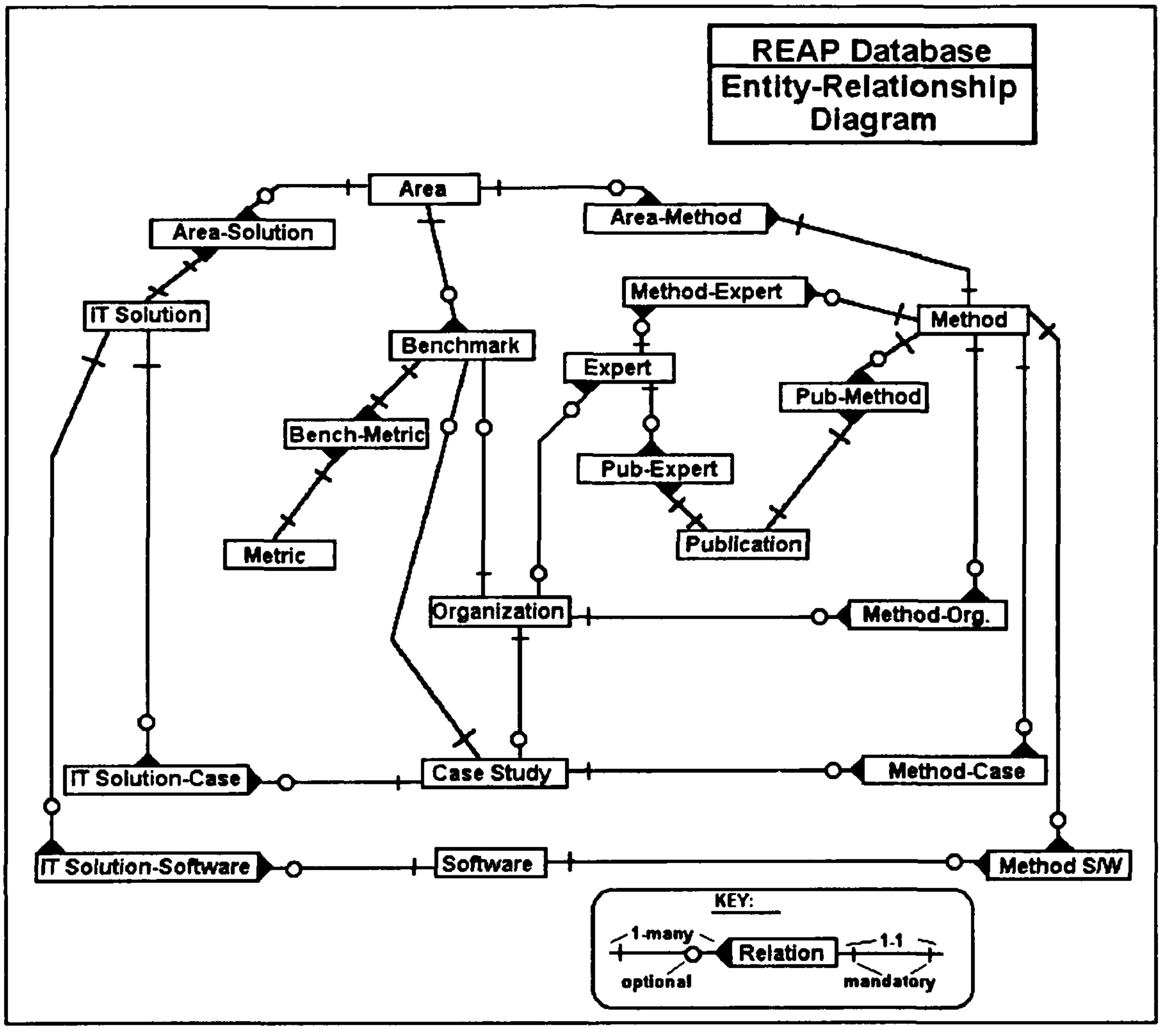


Figure 11

applicable analysis methods, there would not be any Area-Method records, hence the mandatory optional relationship. Six of the relationships are mandatory-mandatory, meaning that for every record that exists in the first relation a related record must exist in the second relation, and for every record that exists in the second relation a related record must exist in the first. This situation also occurs most often with intersection relations. For example, since in the REAP database requirements, every publication must have at least one author, the relationship between Publication and Pub-Expert is

mandatory-mandatory. Every Publication record must have a corresponding Pub-Expert record to link it to the Expert record that contains data about the author while every Pub-Expert record needs a Publication record to exist. The only optional-optional relationship is between Expert and Organization. Neither relation's records needs records in the other to exist. However, should two or more records be associated, the relationship exists.

C. DEVELOPMENT OF THE RELATION DEFINITIONS

The development of the data relation definitions is based on the data object specifications developed in the requirements phase. First ambiguous, non-object property specifications, like "Organization Address" or "Expert Phone", are refined. Second, foreign keys are assigned based on the relationships defined in the Entity Relationship diagram. Finally, the relation are examined to see if they are in domain key normal form.

1. Refinement of Non-Object Properties

An examination of the data objects and object specifications reveals that, despite different names and domains, many properties share common formats. For example, Area Name, Method Name, and Case Name, while describing different things, serve the same function (identifying an object instance) and would have the same format. Many similar properties can be described in this way.

It is also important to distinguish differences in the types of data. The type Character denotes standard alpha-numerics, including upper and lower case letters and

all numbers but excluding punctuation marks. The type Text denotes standard alphanumerics, punctuation, and formatting such as start of paragraph indents and blank lines. The type Numeric denotes decimal numbers. Numerics are distinctive because mathematical operations can be performed on them. It should be noted that, although some attributes consist solely of numbers (e.g. phone number), the Character data type is used because it is nonsensical include them in mathematical operations.

Table 8 illustrates the refinement of the REAP data object properties into data relation attributes.

Table 8.

REAP DATABASE
ATTRIBUTE REFINEMENT DEFINITIONS

Data Object Property	Relation Attribute	Type	Format
Properties Common to More than One Object			
[Object] Name	[Relation]-Name	Character	80 characters long
[xxx] Summary	[xxx]-Sumry	Text	Variable, up to 8,800 characters[1]. Paragraph format
[xxx] Impact [xxx] Result [xxx] Product [xxx] Description	[xxx]-Impact [xxx]-Result [xxx]-Product [xxx]-Descrpt	Text	240 Characters. Sentence format[2].
[xx] Requirements Operating System	[xx]-Req Op-Sys	Text	80 characters long. An item list separated by commas.
[xxx] Phone	Phone Area Code	Character Character	7 digits long 3 digits long
[xxx] Publisher	[xx]-Publisher Phone	Text Character	80 characters long 7 digits long

Table 8.

REAP DATABASE
ATTRIBUTE REFINEMENT DEFINITIONS

Data Object Property	Relation Attribute	Type	Format
Properties Unique to One Object			
Expert Name	Last-Name First-Name MI	Character Character Character	25 characters long 25 characters long 2 characters long
Org. Address	Street City State Zip	Character Character Character Character	80 characters long 40 characters long 2 characters long; all caps 9 digits long
Units	Units	Character	20 characters
Value	Value	Numeric	10 digits (00000000.00)
Year	Year	Character	4 digits long

Notes:
1. Allows five, 22 line pages at 80 characters per line.
2. Allows three 80 character lines.

The final step in the database design is to define the data relations by their attributes. Using the entity relationship diagram, and the attribute definitions as the basis of the design, Pressman's four rules are applied to data relations in order to produce the relation definitions. These relations are listed in Appendix C.

IX. APPLICATION DESIGN

A. APPLICATION DESIGN METHODOLOGY

The second part of Kroenke and Dolan's [1988] design phase is the development of the application design. For online, user-oriented database applications, Kroenke and Dolan [1988] call for an object oriented design method. Pressman [1992] summarizes the object oriented design concept:

> "Object oriented design creates a model ... that can be realized in software. Objects provide a mechanism for representing the information domain, while operations describe the processing that is associated with the information domain. Messages (an interfacing mechanism) provide the means by which operations are invoked."

As stated during the evaluation phase, the application interface compatible with MILNET access is limited to a menu or command line format. A simple command line format would entail the user entering a command or a string of commands upon a prompt by the computer. These commands must be remembered by the user and must be entered correctly. A command/menu interface entails the computer presenting the user with a list of appropriate actions and a corresponding number or letter code which the user enters to activate a particular function. Of the two, a menu format is chosen as simpler for a new user to learn. Pressman [1992] asserts that, "The simple menu provides the user with an overall context and is less error-prone than the command line format."

Kroenke and Dolan [1988] call for a five step application design process:

1. Determine number of applications and application scope.
2. For each application, design control mechanisms that the user will employ to direct the application.
3. For each menu, determine a list of options.
4. For each command and menu option;
   a. Specify the logic
   b. Design materializations
   c. Confirm that database integrity has been maintained.

The implied intent of Kroenke and Dolan's [1988] method is that the specific applications are to be developed as individual, menu-driven objects. These objects are to be designed semi-independently and then brought together into an overall design.

B. DETERMINING THE NUMBER AND SCOPE OF THE APPLICATIONS

The functional specifications are the basis for the application design. An examination of the level 1 data flow diagram (Figure 6) reveals that there are four principle processes, Select Area, Display Methods, Display Benchmarks, and Display IT Solutions. With the exception of Select Area, each was further broken down into sub-processes. Some of these sub-processes were common to two or more principle processes. An examination of the level two data flow diagrams (Figures 7, 8, 9) reveals these sub-processes. The translation of processes and sub-processes into applications is not one-for-one. Redundant sub-processes are represented by a single application that is

called from more than one other application. The functions of other processes or sub-processes can be combined into a single application. Table 9 identifies the applications developed from the functional requirements.

Table 9

REAP DATABASE APPLICATIONS

Application	Scope
Select Area	• Retrieve/Display key fields of all Area records • Facilitate selection of a specific Area record • Display a summary of a selected Area record • Pass Area key field to Display Search Options
Select Search Option	• Facilitate selection of a specific search option (Methods, Benchmarks, or IT Solutions) • Retrieve/Display key fields of related selected option records (Either Method, Benchmark or IT Solution records) • Facilitate selection of a specific record and call to appropriate Display Summary
Display Benchmark Summary	• Retrieve/Display Benchmark record • Retrieve/Display related Bench-Metric records • Facilitate call to Display Case Summary • Facilitate call to Display Organization Summary • Facilitate selection of a Metric record and call to Display Metric Summary
Display Method Summary	• Retrieve/Display a Method record • Retrieve/Display key fields of relate Case Study, Expert, Organization, Publication and Software records • Facilitate selection of a Case Study record and call to Display Case Summary • Facilitate selection of an Expert record and call to Display Expert Summary • Facilitate selection of an Organization record and call to Display Organization Summary • Facilitate selection of a Publication record and call to Display Publication Summary • Facilitate selection of a Software record and call to Display Software Summary

Table 9

REAP DATABASE APPLICATIONS

Application	Scope
Display IT Solution Summary	• Retrieve/Display IT Solution record • Retrieve/Display key fields of related Case Study and Software records • Facilitate selection of a Case Study record and call to Display Case Study • Facilitate selection of a Software record and call to Display Software Summary
Display Expert Summary	• Retrieve/Display Expert record • Facilitate call to Display Organization Summary
Display Organization Summary	• Retrieve/Display Organization record
Display Case Summary	• Retrieve/Display Case Study record • Facilitate call to Display Organization Summary
Display Software Summary	• Retrieve/Display Software record
Display Publication Summary	• Retrieve/Display Publication record • Retrieve/Display key fields of related Expert and Method records • Facilitate selection of Expert record and call to Display Expert Summary • Facilitate selection of Method record and call to Display Method Summary
Display Metric Summary	• Retrieve/Display Metric record

Ten of the eleven applications are taken directly from data flow diagram processes which bear the same name. Six of the applications are called from more than one other applications. The processes Select Method, Select Benchmark, and Select IT Solution were combined into the single application Select Display Option.

C. DESIGN OF USER CONTROL MECHANISMS

An examination of the functional requirements for the various processes along with the scope of the identified applications reveals that the user will control what he/she views by one of three means:

1. Selecting an option or list from a menu screen
2. Selecting a specific record from a list
3. Selecting an option or list from a record summary screen.

1. Selecting an Option from a Menu Screen

In general, a menu screen presents the user with a simple list of options. The user controls the application by selecting the option he/she desires. In the case of the REAP database, the Select Search Option application will be controlled by a menu screen. The user can select the type of information that he/she wishes to view. Menu screens normally allow a user to return to the application that called the menu screen. In the case of Select Search Option, this would mean a return to the Select Area application. Figure 12 provides an example of a menu screen used by the DDN Net Information Center (NIC) database application:

```
Use NIC/Query to access a hierarchy of information about the Defense Data
Network (DDN) and the Network Information Center (NIC) using simple menus.
Bugs to BUG-SERVICE@NIC.DDN.MIL.

**
** Note that a carriage return is required after every command.
** Select menu item 1 for help using this program.
**

  1) HELP -- Introduction, changes, detailed help, help summary.
  2) WHOIS -- Directory of DDN users.
  3) HOSTS -- Describes DDN hosts.
  4) PROTOCOLS -- Describes DDN protocols.
  5) RFCS -- Requests For Comments technical notes.
  6) NIC DOCUMENTS -- Documents available from the NIC.
  7. TACNEWS -- TACnews program.

ROOT: Enter a menu# (1 - 7), or a command ('?' to list).
NIC/Query:
```

Figure 12

2. Selecting a Specific record from a List

As stated in the functional requirements, when more than one record can match a query argument, it is helpful to present the results of a query as a list of key fields from the queried records. The control mechanisms for a list screen allow the user to choose a specific record, view more of the list (if the entire list is too large to fit on one screen), view a previous portion of the list and return to the application that called the list.

3. Selecting an Option from a Summary Screen

An examination of the attributes of the REAP database relations reveals that there are instances where a single attribute may need several screens to be fully displayed. In addition to the specific record being viewed there may be related records or lists of related records to be presented to the user as part of the "summary" of a specific record.

The user needs a simple way to control all these options from any given summary screen. First, the principle record's key field must always appear on the screen to provide a reference for the user. Second, there is a set of standard options that the user can use to control the view of summary. These controls include next screen and previous screen options (if required by the summary size) and a return to the calling application option. Finally, there is a set of options to allow the user to view a related record summary or a list screen of related records. A screen capture of the University of California's MELVYL online library catalog system (Figure 13) provides an example of a summary screen with options:

```
Search request: FIND PA ASIMOV, ISSAC
Search result:  2 records at all libraries

1. Asimov, Isaac, 1920-
    Asimov's Guide to science / by Isaac Asimov.  New York : Basic Books,
  c1972.
      UCB   Astr/Math Q162 .A81 1972
      UCB   Main      Q162 .A81 1972
      UCB   Moffitt   Q162 A82 1972 This library is temporarily closed; see
                 GLADIS for more information.  Some volumes/copies in
                 Moffitt Library. *c2 copies
      UCD   Main Lib  Q162 .A8 1972
      UCD   Main Lib  Q162.A8 1972
      UCD   Phys Sci  Q162.A8 1972
      UCI   Main Lib  Q162 .A8 1972
                         (Record 1 continues on the next screen.)

---------------------------------------------------------------------------
Type choice,  or type HELP for help, END to end session:
  NS  - Next screen of Short display      PA - New Personal Author search
  SHO - Different records in Short        SU - New Subject search
  LON - Long display                      TI - New Title search
  REV - Review display
CAT->
```
Figure 13

D. DEFINITION OF MENUS AND OPTIONS FOR EACH APPLICATION

The purpose of the third step in the design process is to assign specific actions to the applications, defined in the first step, by applying the control mechanisms defined in the second step. The sequence of menus, lists and summaries is then depicted graphically. This graphic is an overview of the entire functionality of the REAP database. Each menu, list or summary depicted is an object in so much as it represents a mechanism for representing the information domain and the associated operations that process the information.

In order to develop a concise design for the REAP database, the issue of how to handle the many times an application requires the user to choose from a list needed to be addressed. A generic list object was created that could be called into use for these situations. Specifically, there are thirteen situations in which the user chooses a specific record from a list. With the exception of the Expert relation, the key fields of the data relations in the REAP database are designed to have identical attributes (Text, 80 characters; see Table 8.) so that a single, standard List application can be used any time a list screen is needed. An application need only call the Standard List application and pass it the query argument and the data relation to be searched. The Standard List application will either return the user's choice for the call to the proper Display Summary application or return without a choice. Using this concept will reduce the amount of functionality required of the Display Summary applications and will simplify the REAP database application design. A materialization of the Standard List Screen concept is provided in Figure 14. In order for the key field information and the list item numbers

```
REAP Database:
Standard List of [relation]
----------------------------------------------------------------------
  1. First key field matching query
  2. Second key field matching query
  3. Third key field matching query
  4. Fourth key field matching query
  5. Fifth key field matching query
  6. Sixth key field matching query
  7. Seventh key field matching query
  8. Eighth key field matching query
  9. Ninth key field matching query
 10. Tenth key field matching query
 11. Eleventh key field matching query
 12. Twelveth key field matching query
 13. Thirteenth key field matching query
 14. Fourteenth key field matching query
----------------------------------------------------------------------
OPTIONS:
(1-999) Number of [relation] to be viewed
 R - Return
NS - Next Screen           PS - Previous Screen

Option ->_
```

Figure 14

to be displayed side by side (as is shown in Figure 14) the key fields will have to be truncated. A truncation of the last five characters of the key field will allow for up to three digit item numbers, a period and a space between the item number and the key field. The key fields of the Expert relation (Last-Name, First-Name, MI) will need to be concatenated into a single field (75 characters long) for use with the standard list. Since this is the only relation for which this procedure is necessary, it should be possible to code the Standard List application to take care of this exception with out significant difficulty.

An overview of the principle objects in the REAP database is presented in Figure 15. This design view reveals the communications between the objects as well as a general idea of the functionality of individual objects. The Standard List object is shown

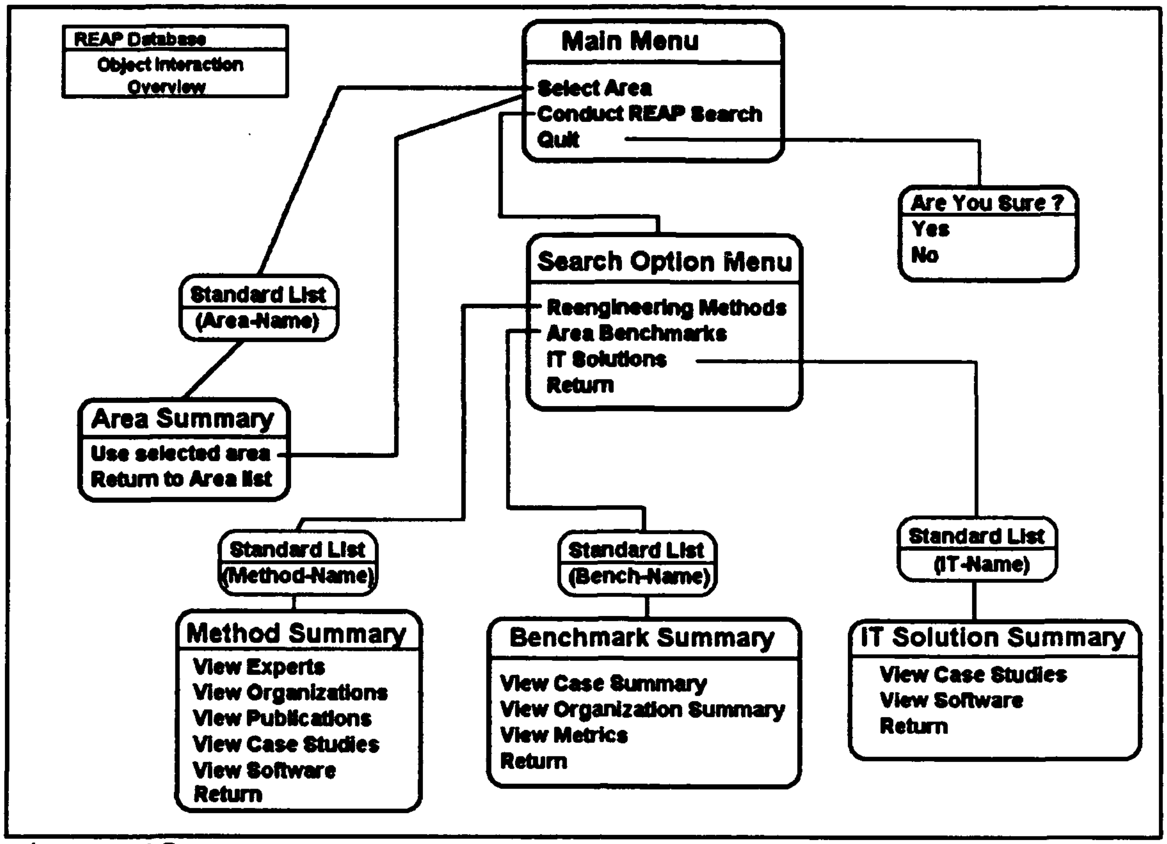


Figure 15

several times in the diagram to illustrate its relationship with the principle objects. In each instance, the key field to be listed for the user is also shown. It should be noted that although it is shown many times, only one object exists and will be coded. The lines emanating from menu or summary options indicate the object activated by the execution of the option. It is understood that the Return option simply returns control to the calling object. One object not directly developed from the functional specifications is the Main Menu object. Its purpose is to coordinate the main activities that the user conducts during a normal database inquiry session. These activities include setting the area filters for the

queries (Select Area), conducting searches of the available data (Conduct REAP Search) and terminating the session (Quit).

Figure 16 provides a more detailed view of the objects that interact with the Method

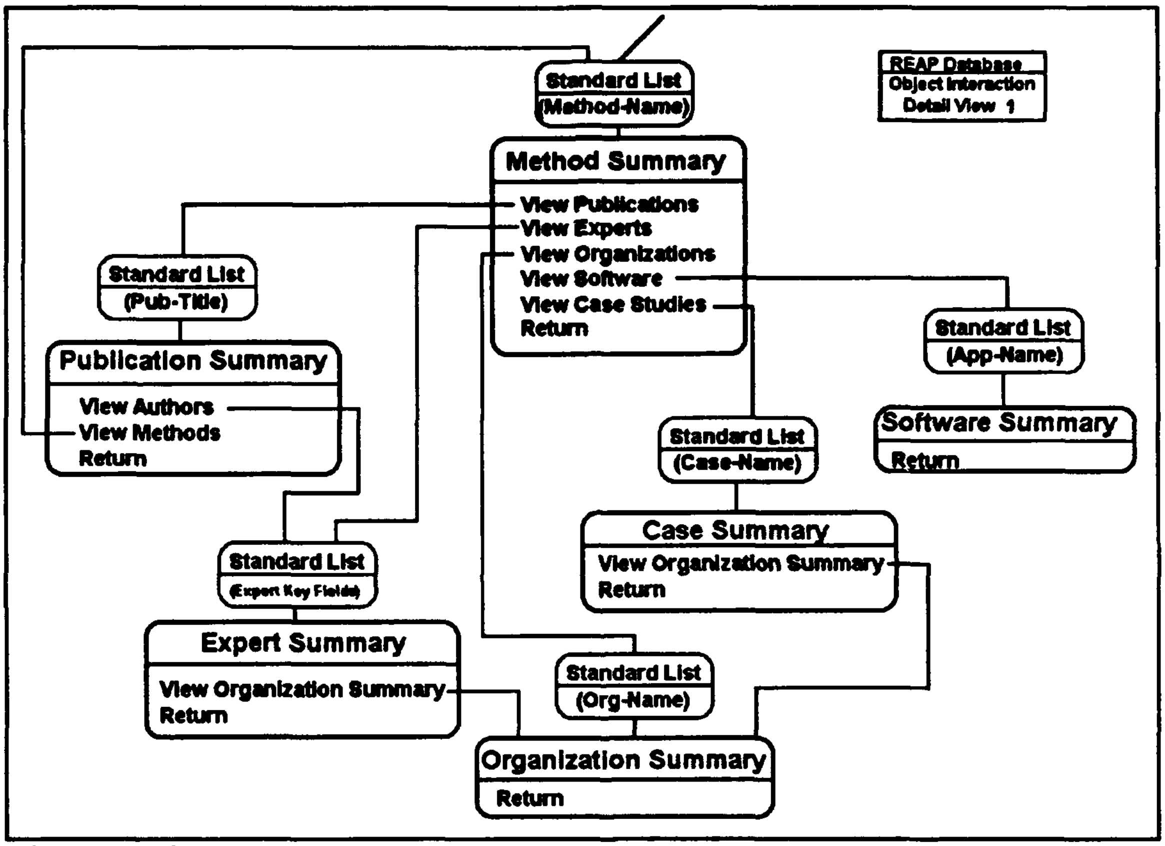


Figure 16

Summary object. The method object is called from the Search Options Menu object. Options provided as part of the method summary allow the user to view lists of records and then summaries of Publications, Experts, Organizations, Case Studies and Software applications. Each of these summary objects can call other summary objects to complete its specific data object view. The Publication Summary object needs to include some type

of control mechanism to limit the number of times a user can cycle through from Method Summary, to Publication Summary, to Method Summary, etc.

Figure 17 provides a detailed view of the interaction between the Benchmark Summary object and the Case Summary, Organization Summary, and Metric Summary

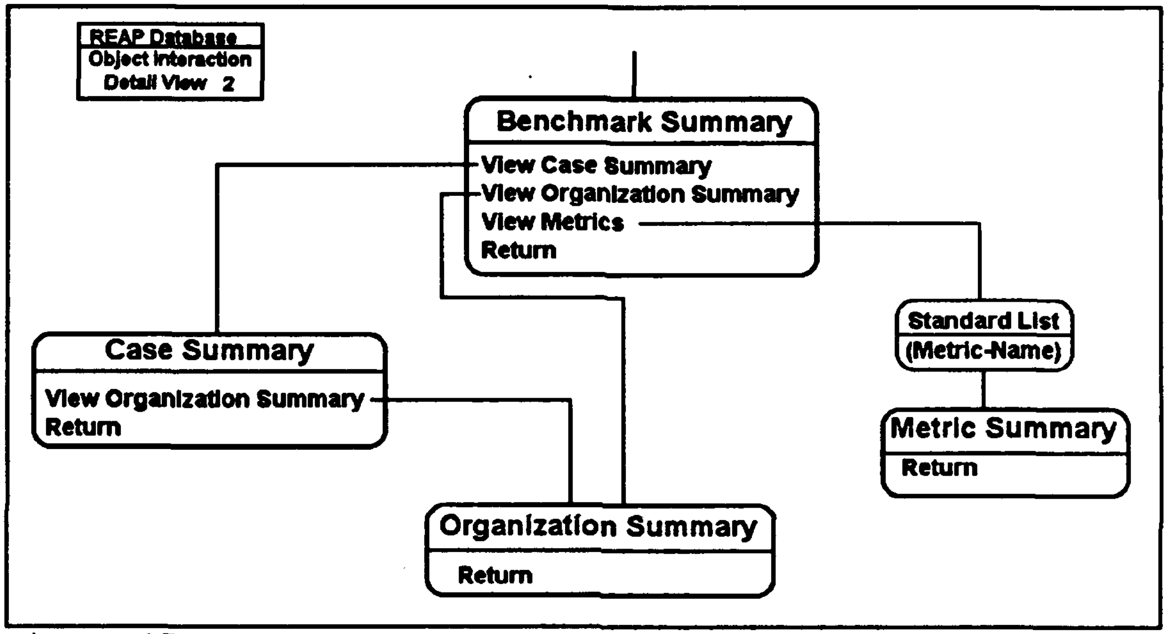


Figure 17

objects. The Benchmark object is called by the Search Option Menu object. There are two ways to activate the Organization Summary object and view information on the Benchmark Organization. The Standard list object is used since there may be more than one metric used to measure with the Benchmark process.

The simplest case of object interaction is found in Figure 18. As before, the IT Solution object is called by the Search Option Menu object. The IT Solution object can call the Software Summary or Case Summary objects via the Standard List object.

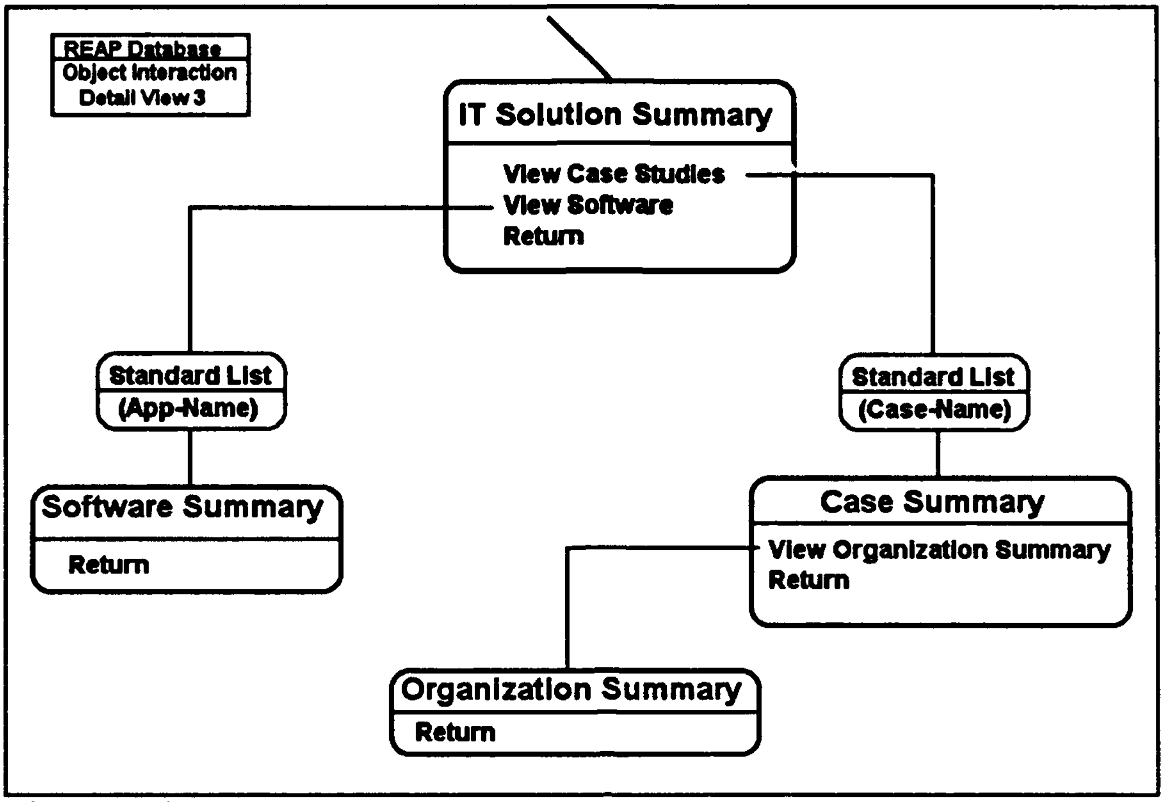


Figure 18

E. APPLICATION LOGIC AND MATERIALIZATIONS

Aside from good software design practice, the major influence on the logic and materialization design is the consideration of human factors. The limits imposed by the TELNET/NVT format do not allow for much innovation with and adaptation of the REAP application interfaces. Screen colors, font, and type size will be dependent on the user's computer system and cannot be controlled by the REAP database application. Additionally, due to differences between character sets employed by various operating systems, the character set used in the REAP database should be limited to either EBCDIC or ANSII standard. These character sets do not contain the special line and box building characters found in the IBM PC character set. ANSII does contain more special

characters (Yen, British pounds, copyright symbol, etc.) than EBCDIC, but these characters will not be used in the user interface. Essentially, the characters used for the user interface design are those that can be produced on standard computer keyboard or typewriter keys (shift and non shift).

Given these limitations, the objective is to design a user interface that is compatible as possible with human physiology and psychology. Pressman, [1992] states that "At the fundamental level, we should understand visual perception, the cognitive psychology of reading, human memory, deductive and inductive reasoning."

Given that users will obtain information from the REAP database by reading screens of text, it is important to consider how humans read when designing the layout of these screens. Hulme [1984] indicates that humans recognize words by their shapes. "In addition to information about letters in the word (or perhaps instead of this) the reader extracts information about what have been called supra-letter features. The most common idea is that the reader uses information about the overall shape of the word." [Hulme, 1984] It is Hulme's assertion that the distribution of ascenders and descenders found in words printed in lower case give it a characteristic outline that is absent when the word is printed in all capitals. It is concluded that all capital words are not as easily recognized and therefore take a longer time to read. An experiment conducted by Tinker that found text printed in all capitals was read about 14% slower than lower case texts.[Hulme, 1984] For this reason, only words that are meant to stand out, such as screen headers or field titles, will be displayed in all capitals. All other information in the REAP database will be stored and displayed in grammatically correct lower case.

An understanding of human inductive and deductive reasoning is important to the design of the commands that the user will use to control the database applications. Pressman [1992] states:

> "Most people do not apply formal inductive or deductive reasoning when confronted with a problem. Rather, we apply a set of heuristics (guidelines, rules, and strategies) based on our understanding of similar problems. In fact, the heuristics that we use tend to be domain specific. That is, an identical problem, encountered in entirely different contexts, might be solved by applying different heuristics. A Human Computer Interface should be specified in a manner that enables the human to develop heuristics for interaction. In general, these heuristics should remain consistent across different interaction domains."

With this in mind, the option command codes are to be consistent for every application in the REAP database. Table 10 summarizes these command codes.

Table 10

REAP DATABASE
OPTION COMMAND CODES

Option	Code	Action
Conduct REAP Search	RS	Calls Search Option Menu
Next Screen	NS	Calls next screen in current display
Previous Screen	PS	Calls previous screen in current display
Quit	Q	Calls termination of connection
Return	R	Returns control to calling object
Review Reengineering and Analysis Methods	RM	Calls standard list of methods
Review Area Benchmarks	AB	Calls standard list of benchmarks
Review IT Solutions	IT	Calls standard list of IT Solutions
Select Area for query	SA	Calls Standard list of Areas
Use selected Area	UA	Returns query argument to main menu

Table 10

REAP DATABASE
OPTION COMMAND CODES

Option	Code	Action
View Case Studies View Case Summary	CS	Calls standard list of Case Studies Calls Case Summary
View Experts View Authors (Experts)	EX	Calls standard list of Experts
View Metrics	ME	Calls standard list of metrics
View Organizations View Organization Summary	OR	Calls standard list of Organizations Calls Organization Summary
View Publications	PU	Calls standard list of Publications
View Software	SW	Calls standard list of Software
Selection from a list	1-999	Retrieves corresponding record

Each command code consists of either one or two letters. The only exception occurs when a user selects a record from a standard list. Then the corresponding item number is entered. No distinction is made between the command for viewing a Case Summary or calling the standard list of case studies. Likewise, no distinction is made between the command for viewing an Organization Summary or calling the standard list of Organizations. This is done in order to simplify the heuristics that the user will need to learn.

Finally, the application logic is specified and the user interface screen materializations are designed. This takes the form of formal object specifications. In these specifications the data relations used by the object are identified, the object's interaction with other objects is described and the logic of its options is defined. The

screen materializations graphically describe all the aspects of all REAP database interfaces. The application object specifications are listed in Appendix D. The screen materializations are contained in Appendix E.

X. PROTOTYPE IMPLEMENTATION

A. OVERVIEW OF IMPLEMENTATION

As previously stated, the functionality of the mainframe-TELNET access prototype is limited to the implementation of the database's data structure. The office of the Director of Defense Information indicated that Oracle would be used to implement the full scale system. Since a version of Oracle that runs on IBM compatible personal computers (Oracle-PC) was available at the Naval Postgraduate School, it was decided that the REAP database prototype would be coded on a on an 33 Mhz, Intel-386 based personal computer using Oracle-PC.

1. Overview of Oracle

Oracle is a relational database management system. Oracle can run on mainframe computers, mini-computers and micro-computers. The system stores information in two-dimensional tables. Each row of a table is a record and each column is an attribute. Oracle uses a high-level query language called Standard Query Language (SQL) to retrieve, modify, insert and delete data in the database. For data retrieval, a SQL statement contains three parts:

Select - Identifies the attributes to be retrieved

From - Identifies the table(s) where the data is stored

Where - Specifies the conditions to be met for retrieval

If data relationships are properly built into the tables, simple SQL statements can be used to retrieve related records. As an example of this simplicity, a SQL statement to retrieve the key fields of the Benchmark records associated with the functional area "Civilian Payroll" is listed below:

```
Select BENCHNAME
From BENCHMARK
Where AREANAME = 'Civilian Payroll';
```

Oracle allows SQL statements to be incorporated as part of its screen design application (SQL*Forms), menu design application (SQL*Menu) and report design application (SQL*Reportwriter). Additionally, SQL statements can be embedded in other programming languages such as C, Fortran and Ada.

Three Oracle datatypes will be used in the REAP database. Character data (CHAR) can be "stored in variable length strings of ASCII or EBCIDIC values." [Dimmick, et al., 1989] String lengths are determined when the table is created. The maximum string length is 240 characters. Numeric data is stored in the NUMBER datatype. The number of digits and decimal places is determined when the table is created. Character attributes that are longer than 240 characters can be stored in the LONG datatype. The LONG datatype can store "variable length character strings up to 65,536 characters." [Dimmick, et al., 1989] Only one LONG attribute is permitted in a table and the LONG attribute cannot be referenced in SQL Where clauses.

For the full scale REAP database, character attributes up to 240 characters (three lines of text at eighty characters per line) will be implemented using the CHAR datatype.

The only numeric attribute is Value, found in the Bench_Metric relation. It will be implemented by the NUMBER datatype with a format of eight digits and two decimal places. All summary (xxx-sumry) attributes will be implemented by the LONG datatype using a string length of 8,800 (five pages at twenty-two lines per page and eighty characters per line).

B. TESTING THE DATA RELATIONS

Experimentation with Oracle-PC revealed that using the LONG datatype to build the prototype database took up so much memory space in the database partition that only part of the data relations could be implemented. Since it is not necessary for the summary attributes to be complete in order to test the data structure, the summary attributes are implemented by 80 or 160 character CHAR datatypes in the prototype. Appendix F lists descriptions of the tables created for the REAP database prototype.

The relations in the REAP database are based on including the key field value of a given record, in a second or more related records. In order to test the database, it was necessary to enter records into the tables created. Sample data collected during the REAP literature search was used. The first line of summaries were used to represent the entire text. In some places where information was incomplete but not critical to the accuracy of the database, sample simulated information was created to fill in the blanks.

In order to verify the REAP database prototype data structure, twenty-six queries were developed. These queries took the form of SQL statements. The purpose of these tests was to determine:

1. Will queries that should return multiple records, such as those that will be used by the standard list object, return the correct list of records?
2. Are the intersection relations properly designed so as to achieve a true many to many relation between tables.
3. Can the correct information be retrieved for a specific summary screen. (This test was especially critical for the Metric summary which must display data from two different tables on the same screen.)

The results of the test queries are found in Appendix G. A slight problem was encountered once when a data entry error caused a slight difference between key field values in two related records. The error was found and corrected but it raised an important design consideration. The data entry mechanism developed should require that a data field be entered into the database only once and only into the primary table (not an intersection table) for that field. The field can then be copied to intersection tables or other related tables. This will ensure that the value of the information is not accidently changed, thus breaking a link between tables.

C. CONCLUSIONS DRAWN FROM THE IMPLEMENTATION PROCESS

The data structure of the REAP database is sound. The query tests confirm that the relations desired in the requirements phase of its development were properly design and

implemented. Oracle's table structure and use of SQL are well suited to the REAP database design and made implementation fairly simple. In all, only about twelve hours were spent creating the tables, populating the prototype database and developing and testing the queries. It could not be determined if the Oracle application tools (SQL*Forms, SQL*Menu, and SQL*Reportwriter) will be adequate for implementation of the user interface. It may be necessary to develop the user interface in Ada and use embedded SQL statements to query the database. The Oracle application tools were used to develop a rudimentary data entry application for the prototype, from which it was determined that they provide the necessary functionality to be used for full scale database administrative applications.

D. RECOMMENDATIONS

During the implementation of the REAP database prototype, several issues surfaced which may enhance quality of support that the database can provide.

Under the present data structure, it is possible to give the user the option to limit the responses to an Method Expert or Method Organization to those instances in his/her geographic area. This could be done by querying on the user's state, area code or zip code as well as the method's name. This feature would allow the user to quickly see consultants or consulting firms that he/she could contact without undue travel expenses.

With a minor data structure change, it would be possible to give the user the option to limit a Benchmark response or Case Study response to organizations that are in a specific branch of the armed forces, the military, or defense industry. In order to do

this, the organization relation would need an organization type attribute. The query would include this attribute as part of the Where clause. This feature would allow a user to view Benchmarks or case studies of organizations that are not too dissimilar to their own.

Under the present data structure, it is possible to give the user the option to view all the Expert instances employed by a given organization. This would be useful if a user knew where an individual worked but did not know the correct spelling of his name.

It would be possible to allow a user to limit the responses to a Software query to applications that are compatible with his/her computer system. Since many software publishers produce versions of the same application that run on different computer systems, a data relation called Hardware and an intersection relation Hardware_Software would need to be created. These relations would establish the many to many relationship that can exist between Software applications and the systems that support them. Since this would be more than a minor change in the database structure, it is recommended that it be implemented only if user feedback indicates it is desired.

Finally, it is recommended that a fairly broad interpretation is used when linking Method records and IT Solution records with Area records. The determination should be made on whether there is a possibility that method or solution would apply to a specific area and not on the likelihood that it will apply. While the stated intent of the REAP database is to provide information specific to a functional area, it is felt that it is better to include a little more information than is needed rather than a little less.

APPENDIX A. DATA OBJECTS

The formal data object specifications are listed below in alphabetical order.

Area Object:

Area Name	-Text; Name of business area (i.e. payroll, inventory control, etc)
Area Description	-Text; Explanation of the area's purpose, objectives
Benchmark	-Benchmark object; Benchmark for business area
Method	-Method Object; Analysis methods applicable to the business area
IT Solution	-IT Solution object; IT solutions applicable to the business area

Benchmark Object:

Benchmark Name	-Text; Name of benchmark
Value	-Numeric; Quantity of benchmark units
Organization	-Organization object; Description of benchmark organization
Case Study	-Case Study object; Applicable case study of benchmark process
Metric	-Metric object; Metric associated with benchmark
Process Summary	-Text(long); Summary of the benchmark process
Area	-Area object; Area associated with Benchmark

Case Study Object:

Case Name	-Text; Name of case study
Organization	-Organization object; Organization that the case study examines
Case Summary	-Text (long); Description of case study, findings, etc.
IT Solution	-IT Solution object; IT Solution(s) related to the Case Study
Method	-Method object; Method(s) illustrated in the case study
Benchmark	-Benchmark object; Benchmark process illustrated in the case study

Expert Object:

Name	-Text; Expert's Name and title
Organization	-Organization Object; Organization Expert is affiliated with
Address	-Text; Expert's address (specific to expert vice organization)
Phone	-Character; Expert's phone number
Position	-Text; Position that expert hold in affiliated organization.
Publication	-Publication object; Any publications authored by the expert
Method	-Method object; Method(s) that the expert practices

IT Solution Object:

Solution Name	-Text; General name for solution method (i.e. Networking)
IT Summary	-Text(long); Brief explication of solution
Sys. Requirements	-Text; Generic list of resources (Hardware types, communications requirements, training personnel, etc) needed to implement solution

IT Impact	-Text; List of results commonly experienced as result of implementation of solution
Case Study	-Case Study object; Applicable, related case studies
Software	-Software object; Computer applications that can be used to implement the IT solution
Area	-Area object; Areas associated with IT Solution

<u>Method Object</u>:

Method Name	-Text; Name/title of method
Summary	-Text(long); Outline of what the method does/how it works
Method Result	-Text; description of output/benefits of implementing method
Case Study	-Case Study object; Case studies related to method
Expert	-Expert object; Experts involved with the method
Publication	-Publication object; Publication(s) related to/describing method
Organization	-Organization object; Organization(s) involved with implementing the method
Area	-Area object; Areas associated with Method

<u>Metric Object</u>:

Metric Name	-Text; Name of metric
Use	-Text(long); Explanation of how the metric is applied
Units	-Text; Specification of units of measure for the metric (i.e. man-hours, dollars, percentage increase in output, etc)
Benchmark	-Benchmark object; Benchmark(s) for which the metric is used

Organization Object:

Organization Name	-Text; Name of organization(to include parent organization; i.e. NS Norfolk Supply Depot, US Navy)
Org. Address	-Text; Full mailing address
Org. Products	-Text; Description of Organization's output (i.e. payroll for 1500 workers)
Org. Description	-Text(long); Summary of what the organization does (i.e. process civilian pay and personnel records, calculates correct amount of wages due based on hours worked, tax withholding, etc)
Org. Phone	-Character; Contact point phone number
Method	-Method object; Method(s) that the organization practices
Software	-Software object; Software applications produced by organization
Expert	-Expert object; Experts employed by the organization

Publication Object:

Title	-Text; Title of publication to include periodical references
Expert	-Expert object; Author of the publication
Method	-Method object; Method(s) described in the publication
Publisher	-Text; Name/location of the publisher
Year	-Numeric(4 digits); Year published
Pub. Summary	-Text(long); Summarization of main points made in the publication

Software Object:

Application Name	-Text; Name of software application (to include version number)
Organization	-Organization object; Organization that produces the application
Operating System	-Text; List of operating systems that support the application
H/W Requirements	-Text; List of hardware requirements for the application to run.
S/W Description	-Text; Description of use/benefits of the application.
IT Solution	-IT Solution object; IT Solution implemented by the software
Method	-Method object; Method supported by the software

APPENDIX B. FUNCTIONAL SPECIFICATIONS

<u>Select Area</u>

- **Output Description:**
  - List of all AREA instances in the REAP DB
  - Description of a selected Area instance (optional)
- **Source Data:**
  - AREA object
- **Processing Notes:**
  - User area input necessary to limit search to manageable size
  - Used to select AREA filter for all subsequent queries
- **Volume:**
  - One to three times per use
  - Unknown number of uses per day
- **Frequency:**
  - Daily

<u>Display Benchmarks</u>

- **Output Description:**
  - List of BENCHMARK instances related to selected AREA instance

- Screen showing summary of the selected area's BENCHMARK instance
- Screen showing summary of corresponding CASE STUDY instance
- Screen showing summary of corresponding ORGANIZATION instance
- List of corresponding METRIC instances
- Screen showing summary of a selected METRIC instance

o **Source Data:**

- BENCHMARK object
- CASE STUDY object
- ORGANIZATION object
- METRIC object

o **Processing Notes:**

- Initial screen showing benchmark summary provides options to select Case Study summary screen, Organization summary screen and Metrics list.
- Metrics list allows user to select desired metric summary
- Allow for return to Benchmark summary screen from all sub screens.

o **Volume:**

- Same as or slightly less than Select Area

o **Frequency:**

- Daily

<u>Display Solutions</u>

o **Output Description:**

- List of IT SOLUTION instances related to selected AREA instance
- Screen showing summary of selected IT SOLUTION instance
- List of CASE STUDY instances related to selected IT SOLUTION instance
- List of SOFTWARE instances related to selected IT SOLUTION instance
- Screen showing a summary of the selected CASE STUDY instance to include an optional screen showing a summary of the ORGANIZATION instance mentioned in selected CASE STUDY
- Screen showing a summary of the selected SOFTWARE instance to include an optional screen showing a summary of the SOFTWARE's publisher (an ORGANIZATION instance)

o **Source Data:**

- IT SOLUTION object
- CASE STUDY object
- SOFTWARE object
- ORGANIZATION object

o **Processing Notes:**

- Initial Solutions screen shows list of IT Solutions from which a Solution summary is selected for viewing
- Solution summary screen allows viewing lists of related software applications and case studies; Case and Software summaries are selected from these lists
- Case Study and Software summary screens allow the viewing of a related organization description

- Allow return to Solution summary screen from Case and Software summary screens
- Allow return to IT Solution list from Solution summary screen

o **Volume:**

- Several times per use

o **Frequency:**

- Daily

Display Methods

o **Output Description:**

- List of METHOD instances related to selected AREA instance (or PUBLICATION instance)
- Screen showing a summary of the selected METHOD instance
- List of CASE STUDY instances related to selected METHOD instance
- List of EXPERT instances related to selected METHOD instance
- List of PUBLICATION instances related to selected METHOD instance
- List of ORGANIZATION instances related to selected METHOD instance
- List of SOFTWARE instances related to selected METHOD instance
- Screen showing a summary of the selected CASE STUDY instance to include an optional screen showing a summary of the ORGANIZATION instance mentioned in selected CASE STUDY

- Screen showing a summary of the selected SOFTWARE instance to include an optional screen showing a summary of the SOFTWARE's publisher (an ORGANIZATION instance)
- Screen showing a summary of the selected EXPERT instance to include an optional screen showing a summary of the expert's employer (ORGANIZATION instance)
- Screen showing a summary of the selected PUBLICATION instance to include an optional "About the author" screen (EXPERT instance)
- Screen showing a summary of the selected ORGANIZATION instance

o **Source Data:**

- METHOD object
- CASE STUDY object
- SOFTWARE object
- ORGANIZATION object
- EXPERT object
- PUBLICATION object

o **Processing Notes:**

- Initial Methods screen shows list of applicable Methods from which a Method summary is selected for viewing
- Method summary screen allows viewing lists of related experts, organizations, publications, software applications and case studies; summaries are selected from these lists

- Case Study, Expert and Software summary screens allow the viewing of a related organization summary screen
- Publication summary screen allows the viewing of a related (author of publication) Expert summary screen(s) and a list of other METHOD instances covered in the publication.
- Allow return to Method summary screen from all other secondary summary and list screens
- Allow return to Methods list from Method summary screen
- It may be necessary to limit the number of iterations that it is possible to "circle back" to the initial Methods list via the Publication summary.

o **Volume:**

- Several times per use

o **Frequency:**

- Daily

APPENDIX C. DATA RELATIONS

Key attributes are underlined and foreign key attributes denoted with an asterisk (*)

Principle Relations:

AREA

Area-Name | A-Descrpt

BENCHMARK

Bench-Name | Process_Sumry | Org-Name* | Case-Name* | Area-Name*

IT SOLUTION

IT-Name | IT-Sumry | Sys-Req | Impact

METHOD

Meth-Name | M-Sumry | M-Result

Secondary Relations:

CASE STUDY

<u>Case-Name</u> | Case-Sumry | Org-Name*

EXPERT

<u>Last-Name</u> | <u>First-Name</u> | <u>MI</u> | Position | Area-Code | Phone | Org-Name*

METRIC

<u>Metric-Name</u> | Use-Descrpt | Units

ORGANIZATION

<u>Org-Name</u> | Street | City | State | Zip | Area-Code | Phone | Org-Product | Org-Descrpt

PUBLICATION

<u>Pub-Title</u> | Publisher | Year | Pub-Sumry | Area-Code | Phone

SOFTWARE

<u>App-Name</u> | Op-Sys | HW-Req | SW-Descrpt | SW-Publisher | Phone | Area-Code

Intersection Relations:

AREA-METHOD

<u>Area-Name*</u> | <u>Meth-Name*</u>

AREA-IT SOLUTION

<u>Area-Name*</u> | <u>IT-Name*</u>

BENCH-METRIC

<u>Bench-Name*</u> | <u>Metric-Name*</u> | Value

IT SOLUTION-CASE

<u>IT-Name*</u> | <u>Case-Name*</u>

IT SOLUTION-S/W

<u>IT-Name*</u> | <u>App-Name*</u>

METHOD-EXPERT

<u>Meth-Name*</u> | <u>Last-Name*</u> | <u>First-Name*</u> | <u>MI*</u>

METHOD-S/W

<u>Meth-Name*</u> | <u>App-Name*</u>

METHOD-ORG.

<u>Meth-Name*</u> | <u>Org-Name*</u>

METHOD-PUB

<u>Meth-Name*</u> | <u>Pub-Title*</u>

METHOD-CASE

<u>Meth-Name</u>* | <u>Case-Name</u>*

PUB-EXPERT

<u>Pub-Title</u>* | <u>Last-Name</u>* | <u>First-Name</u>* | <u>MI</u>*

APPENDIX D. OBJECT DESIGN SPECIFICATIONS

OBJECT NAME:
Main Menu
RELATIONS USED:
None
OBJECTS CALLED:

Name:	**Via**	**Data Passed**
Area Summary	Standard List	None
Search Option Menu	Direct	Area-Name
Are You Sure?	Direct	None

DISPLAYS:

Screen:	**Information:**
1	REAP Database header, Main Menu header and Main Menu Options. (One screen possible)

OPTIONS:

1. Select Area:
   Call Area Summary. Receive either an instance of Area-Name or a null value.

2. Conduct REAP Search:
   Call Search Option Menu. Pass Area-Name. No values returned.

3. Quit
   Call Are You Sure? If Yes value returned, program terminates user connection. If No value returned, maintain user connection.

NOTES:

1. Select Area option must be executed and a non-null value for Area-Name must be obtained before Conduct REAP Search option can be executed.

OBJECT NAME:
Area Summary
RELATIONS USED:
Area
OBJECTS CALLED:

Name:	Via	Data Passed
None		

CALLED BY:

Name:	Via	Data Passed
Standard List	--	Area-Name

DISPLAYS:

Screen:	Information:
1	REAP Database header, Area-Name, Description of Area (Area-Descrpt), and Options. (One screen possible)

OPTIONS:

1. Use Selected Area
   Return selected Area-Name instance to Main Menu. (Selected Area-Name instance to be used as query argument for subsequent REAP searches.)

2. Return to Area List
   Return to Standard List

NOTES:

1. If Use Selected Area option activated, control passes directly back to Main menu.

OBJECT NAME:
Search Option Menu

RELATIONS USED:
Area

OBJECTS CALLED:

Name:	Via	Data Passed
Method Summary	Standard List	Area-Name
Benchmark Summary	Standard List	Area-Name
IT Solution Summary	Standard List	Area-Name

CALLED BY:

Name:	Via	Data Passed
Main Menu	Direct	Area-Name

DISPLAYS:

Screen:	Information:
1	REAP Header, Search Option Menu Header, and Menu Options. (One screen possible)

OPTIONS:

1. Review Reengineering/Analysis Methods (RM)
   Call Method Summary via Standard List. Pass Area-Name. No values returned.

2. Review Area Benchmarks (BE)
   Call Benchmark Summary via Standard List. Pass Area-Name. No values returned.

3. Review IT Solutions (IT)
   Call Benchmark Summary via Standard List. Pass Area-Name. No values returned.

4. Return
   Return control to Main Menu

OBJECT NAME:
Standard List

RELATIONS USED:

Area	Area-Method	Area-IT Solution
Bench-Metric	IT Solution-Case	IT Solution-S/W
Method-Expert	Method-S/W	Method-Org
Method-Pub	Method-Case	Pub-Expert

OBJECTS CALLED:

Name:	Via	Data Passed
Area Summary	--	Area-Name
Method Summary	--	Method-Name
Benchmark Summary	--	Bench-Name
IT Solution Summary	--	IT-Name
Publication Summary	--	Pub-Title
Expert Summary	--	Last-Name,First-Name, MI
Case Summary	--	Case-Name
Software Summary	--	App-Name
Metric Summary	--	Metric-Name, Value

CALLED BY:

Name:	Via	Data Passed
Main Menu	--	None
Search Option Menu	--	Area-Name
Method Summary	--	Method-Name
Benchmark Summary	--	Bench-Name
IT Solution Summary	--	IT-Name
Publication Summary	--	Pub-Title

DISPLAYS:

Screen:	Information:
1	REAP Database header, Standard List header (use name of relation in list header, e.g. "List of Areas"). Fourteen (14) lines of listed key fields (truncated to 75 characters) with three digit leading item numbers. Standard List options. (Many screens possible)

OPTIONS:

1. Any valid list item number (1-999)
   Call the appropriate summary object. Pass the value of the key field(s) selected.

2. Next Screen (NS)

Display the key fields of the next 14 records (counted from the last record displayed on the current screen) or remaining records if less than 14, that match the query argument. Re-display all headers and options. Re-display the last screen if this option is selected from the last screen.

3. Previous Screen (PS)
   Display the key fields of the previous 14 records (counted from the first record of the current screen) that match the query argument. Re-display all headers and options. Re-display the first screen if this option is activated from the first screen.

4. Return
   Return to calling object. No values returned.

NOTES:

1. The Expert key fields Last-Name, First-Name, MI must be concatenated into a 75 character long string in order to be presented in a list format. Included in the 75 characters are comas and spaces between names. Last-Name cut to 38 characters max, first name cut to 37 characters max, MI stays at one character. Given the length of most American names, this should not pose a problem.

OBJECT NAME:
Method Summary

RELATIONS USED:
Method   Area-Method

OBJECTS CALLED:

Name:	Via	Data Passed
Publication Summary	Standard List	Meth-Name
Expert Summary	Standard List	Meth-Name
Organization Summary	Standard List	Meth-Name
Case Summary	Standard List	Meth-Name
Software Summary	Standard List	Meth-Name

CALLED BY:

Name:	Via	Data Passed
Search Option Menu	Standard List	Meth-Name

DISPLAYS:

Screen:	Information:
1	Method Summary header (include Meth-Name), Method Results(M-Result), first five lines of Summary (M-Sumry) and Options. (One screen possible)
2	Method Summary header, 16 lines of M-sumry and Options. (Up to seven screens possible; total of 8)

OPTIONS:

1. View Publications (PU)
   Call Publication Summary via Standard List. Pass Meth-Name. No values returned.

2. View Experts (EX)
   Call Expert Summary via Standard List. Pass Meth-Name. No values returned.

3. View Case Studies (CS)
   Call Case Summary via Standard List. Pass Meth-Name. No values returned.

4. View Organizations (OR)
   Call Organization Summary via Standard List. Pass Meth-Name. No values returned.

5. View Software (SW)
   Call Software Summary via Standard List. Pass Meth-Name. No values returned.

6. Return (R)
Return control to Standard List (Methods)

7. Next Screen (NS)
Available for first thru seventh screen. Retrieve next 16 lines of M-Sumry and display using a screen 2 format.

8. Previous Screen (PS)
Available for second thru eighth screen. If third thru eighth screen, retrieve previous 16 lines of M-Sumry and display using a screen 2 format, else (second screen) retrieve first five lines of M-Sumry and display using a screen 1 format.

OBJECT NAME:
Benchmark Summary

RELATIONS USED:
Benchmark

OBJECTS CALLED:

Name:	Via	Data Passed
Case Summary	Direct	Case-Name
Organization Summary	Direct	Org-Name
Metric Summary	Standard List	Bench-Name

CALLED BY:

Name:	Via	Data Passed
Search Option Menu	Standard List	Bench-Name

DISPLAYS:

Screen:	Information:
1	Benchmark summary header (include Area-Name and Bench-Name), Benchmark organization (Org-Name), eight lines of Process-sumry, and options. (One screen possible)
2	Benchmark summary header, 16 lines of Process-Sumry and options. (Up to seven screens possible)

OPTIONS:

1. View Case Summary (CS)
   Call Case Summary. Pass Case-Name. No values returned.

2. View Organization Summary (OR)
   Call Organization Summary. Pass Org-Name. No values returned.

3. View Metrics (ME)
   Call Metrics Summary via Standard List. Pass Bench-Name. No values returned.

4. Return (R)
   Return control to Standard List (Benchmarks).

5. Next Screen (NS)
Available for first thru seventh screen. Retrieve next 16 lines of Process-Sumry and display using a screen 2 format.

6. Previous Screen (PS)
Available for second thru eighth screen. If third thru eighth screen, retrieve previous 16 lines of Process-Sumry and display using a screen 2 format, else (second screen) retrieve first eight lines of Process-Sumry and display using a screen 1 format.

OBJECT NAME:
IT Solution Summary

RELATIONS USED:
IT Solution

OBJECTS CALLED:

Name:	Via	Data Passed
Software Summary	Standard List	IT-Name
Case Summary	Standard List	IT-Name

CALLED BY:

Name:	Via	Data Passed
Search Option Menu	Standard List	IT-Name

DISPLAYS:

Screen:	Information:
1	IT Solution summary header (include IT-Name), System Requirements (Sys-Req), IT Impact (Impact) and options. (One screen possible)
2	IT Solution summary header, 16 lines of IT-Sumry and options. (Up to seven screens possible)

OPTIONS:

1. View Case Studies (CS)
   Call Case Summary via Standard List. Pass IT-Name. No values returned.

2. View Software (SW)
   Call Software Summary via Standard List. Pass IT-Name. No values returned.

3. Return (R)
   Return control to Standard List (IT Solutions)

4. Next Screen (NS)
   Available for first thru seventh screen. Retrieve next 16 lines of IT-Sumry and display using a screen 2 format.

5. Previous Screen (PS)
   Available for second thru eighth screen. If third thru eighth screen, retrieve previous 16 lines of M-Sumry and display using a screen 2 format, else (second screen) display using a screen 1 format.

OBJECT NAME:
Publication Summary

RELATIONS USED:
Publication

OBJECTS CALLED:

Name:	Via	Data Passed
Method Summary	Standard List	Pub-Title
Expert Summary	Standard List	Pub-Title

CALLED BY:

Name:	Via	Data Passed
Method Summary	Standard list	Pub-Title

DISPLAYS:

Screen:	Information:
1	Publication Summary header (include Pub-Title), Publisher (Name and Phone), year published (Year), first eight lines of Publication Summary (Pub-Sumry) and options. (one screen possible
2	Publication Summary header, 16 lines of Pub-Sumry and options. (Up to seven screens possible)

OPTIONS:

1. View other Reengineering/Analysis Methods covered (RM)
   Call Method Summary via Standard List. Pass Pub-Title. No values returned.

2. View Authors (Expert) (EX)
   Call Expert Summary via Standard List. Pass Pub-Title. No values returned.

3. Return (R)
   Return control to Standard List (Publications)

4. Next Screen (NS)
   Available for first thru seventh screen. Retrieve next 16 lines of Pub-Sumry and display using a screen 2 format.

5. Previous Screen (PS)
   Available for second thru eighth screen. If third thru eighth screen, retrieve previous 16 lines of Pub-Sumry and display using a screen 2 format, else (second screen) retrieve first eight lines of Pub-Sumry and display using a screen 1 format.

OBJECT NAME:
Expert Summary

RELATIONS USED:
Expert

OBJECTS CALLED:

Name:	Via	Data Passed
Organization Summary	Direct	Org-Name

CALLED BY:

Name:	Via	Data Passed
Method Summary	Standard List	Last-Name, First-Name, MI
Publication Summary	Standard List	Last-Name, First-Name, MI

DISPLAYS:

Screen:	Information:
1	Expert Summary Header, Last Name, First Name, MI, Position, Organization Name (Org-Name), Full phone number (Area-Code and Phone) and options. (One screen possible)

OPTIONS:

1. View Organization Summary (OR)
   Call Organization Summary. Pass Org-Name. No values returned.

2. Return (R)
   Return control to Standard List (Experts)

OBJECT NAME:
Organization Summary

RELATIONS USED:
Organization

OBJECTS CALLED:

Name:	Via	Data Passed
None		

CALLED BY:

Name:	Via	Data Passed
Method Summary	Standard List	Org-Name
Expert Summary	Direct	Org-Name
Case Summary	Direct	Org-Name
Benchmark Summary	Direct	Org-Name

DISPLAYS:

Screen:	Information:
1	Organization Header (include Org-Name), Organization Address (Street, City, State, Zip), Organization phone number (Area-Code, Phone), Organization Product Description (Org-Product) and Organization Description (Org-Descrpt) (One screen possible)

OPTIONS:

1. Return (R)

   Return control to calling object.

OBJECT NAME:
Case Summary

RELATIONS USED:
Case Study

OBJECTS CALLED:

Name:	Via	Data Passed
Organization Summary	Direct	Org-Name

CALLED BY:

Name:	Via	Data Passed
Method Summary	Standard List	Case-Name
Benchmark Summary	Direct	Case-Name
IT Solution Summary	Standard List	Case-Name

DISPLAYS:

Screen:	Information:
1	Case Summary header (include Case-Name), Subject Organization (Org-Name), first ten lines of the case summary(Case-Sumry) and options. (One screen possible)
2	Case Summary header, 16 lines of Case-Sumry and options. (Seven screens possible)

OPTIONS:

1. View Organization Summary (OR)
   Call Organization Summary. Pass Org-Name. No values returned.

2. Return (R)
   Return control to calling object.

4. Next Screen (NS)
   Available for first thru seventh screen. Retrieve next 16 lines of Case-Sumry and display using a screen 2 format.

5. Previous Screen (PS)
   Available for second thru eighth screen. If third thru eighth screen, retrieve previous 16 lines of Case-Sumry and display using a screen 2 format, else (second screen) retrieve first ten lines of Case-Sumry and display using a screen 1 format.

OBJECT NAME:
Software Summary

RELATIONS USED:
Software

OBJECTS CALLED:

Name:	Via	Data Passed
None		

CALLED BY:

Name:	Via	Data Passed
Method Summary	Standard list	App-Name
IT Solution Summary	Standard List	App-Name

DISPLAYS:

Screen:	Information:
1	Software Summary header(include App-Name), Operating System requirements (OP-Sys), Hardware requirements (H/W-Req), Software Publisher (S/W-Publisher, Phone, Area-Code), Software description (S/W-Descrpt), and options (One screen possible)

OPTIONS:

1. Return (R)
   Return control to Standard List (Software)

OBJECT NAME:
Metric Summary

RELATIONS USED:
Metric Bench-Metric

OBJECTS CALLED:

Name:	Via	Data Passed
None		

CALLED BY:

Name:	Via	Data Passed
Benchmark Summary	Standard List	Bench-Name, Value, Metric-Name

DISPLAYS:

Screen:	Information:
1	Benchmark Summary Header (include Bench-Name), Metric Identification (Metric-Name), Description of Use (Use-Descrpt), Measure of Benchmark (Value, Units), and Options. (One screen possible)

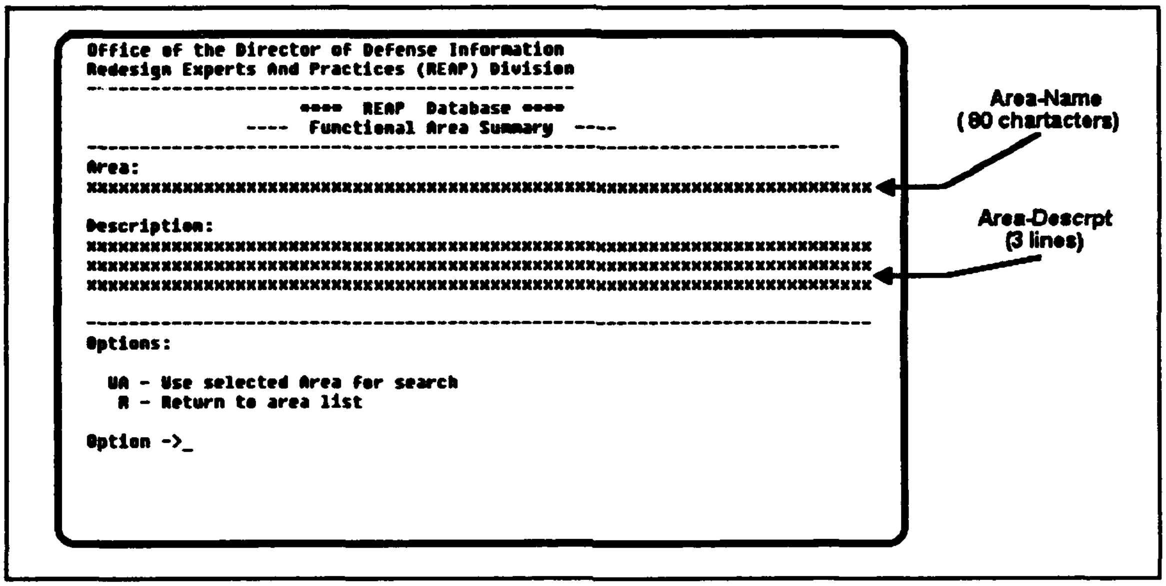
OPTIONS:
1. Return (R)
   Return control to Standard List (Metrics)

NOTES:
None

APPENDIX E. SCREEN MATERIALIZATIONS

```
Office of the Director of Defense Information
Redesign Experts And Practices (REAP) Division
----------------------------------------------

                    **** REAP  Database ****
                      ---- Main Menu  ----


-----------------------------------------------------------------------
OPTIONS:

                  SA  -  Select Area for query

                  RS  -  conduct REAP Search

                   Q  -  Quit REAP Database

-----------------------------------------------------------------------
Enter one option

Option -> _
```

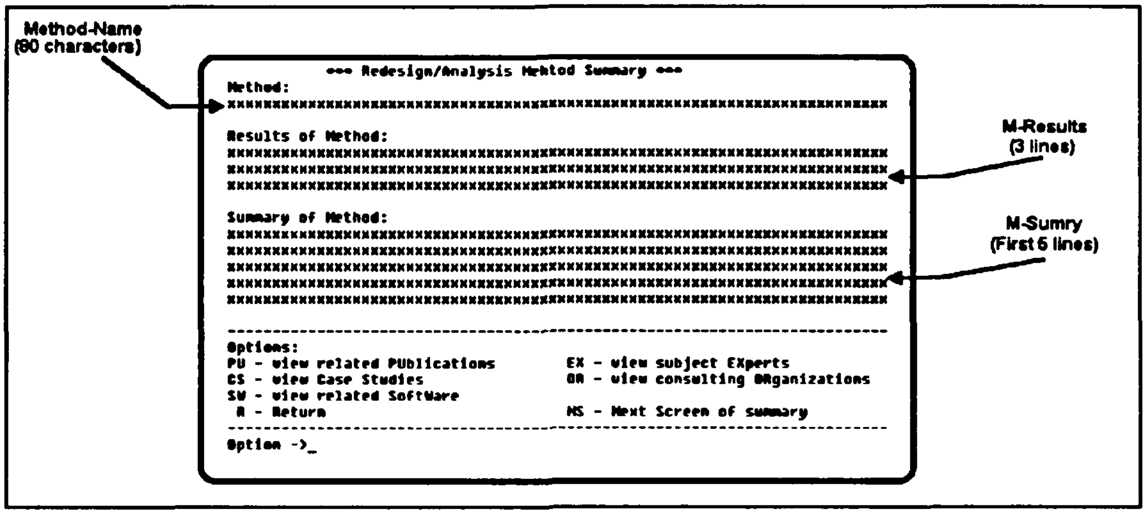
Main Menu screen



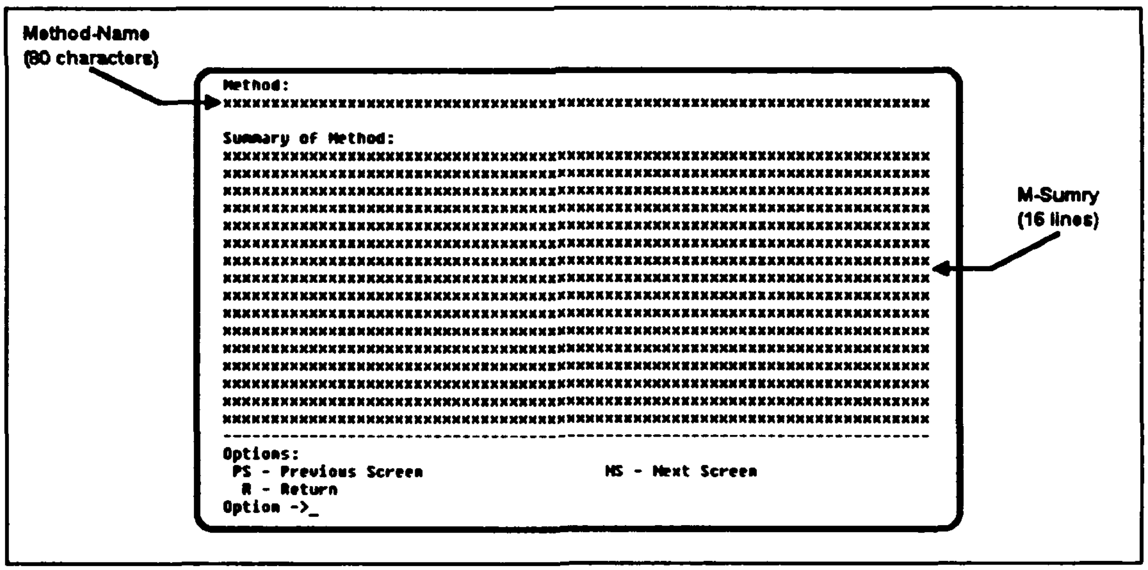
Area Summary screen

```
Office of the Director of Defense Information
Redesign Experts And Practices (REAP) Division
----------------------------------------------
                    **** REAP  Database ****
                 ----  REAP Search Options  ----
-----------------------------------------------------------------------
Select one option

Options:

  RM - review Redesign/analysis Methods

  BE - review area BEnchmarks

  IT - review Information Technology solutions

   R - Return to main menu
  ------------------------------------------------------------------------
Option ->_
```

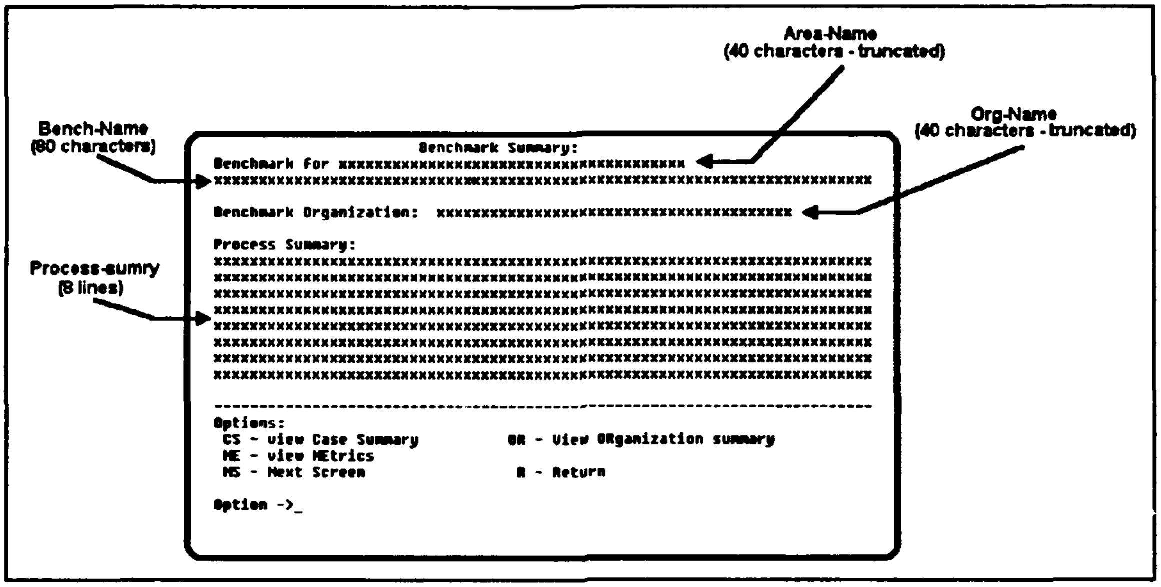
REAP Search Options Menu screen



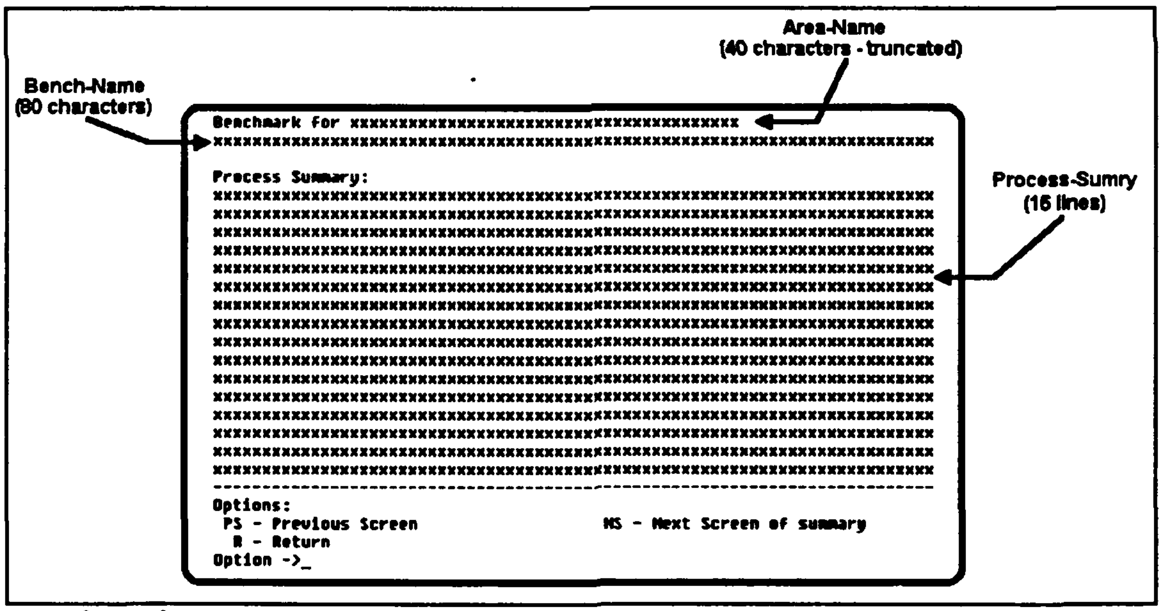
Method Summary screen 1



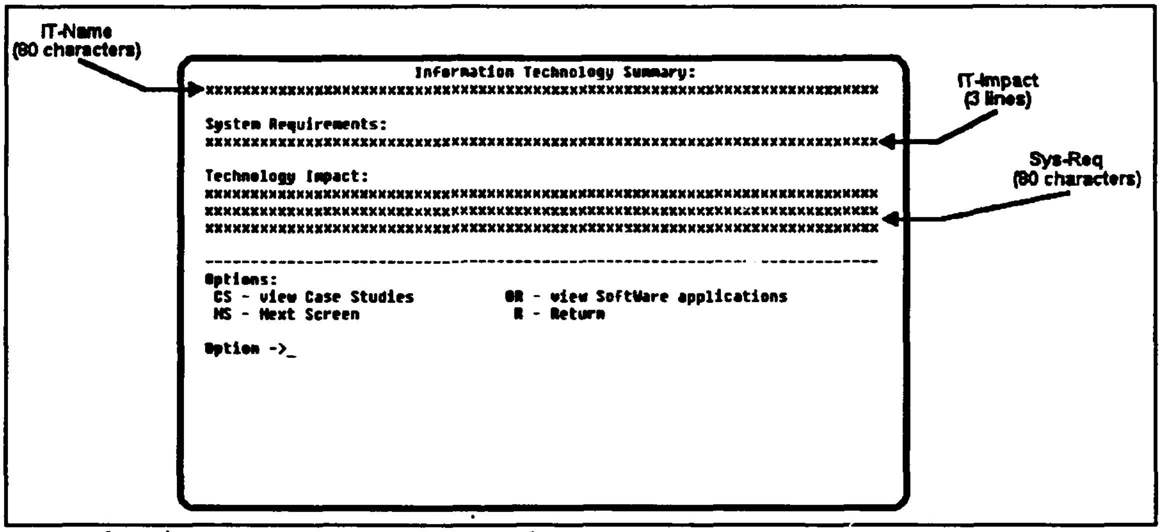
Method Summary screen 2



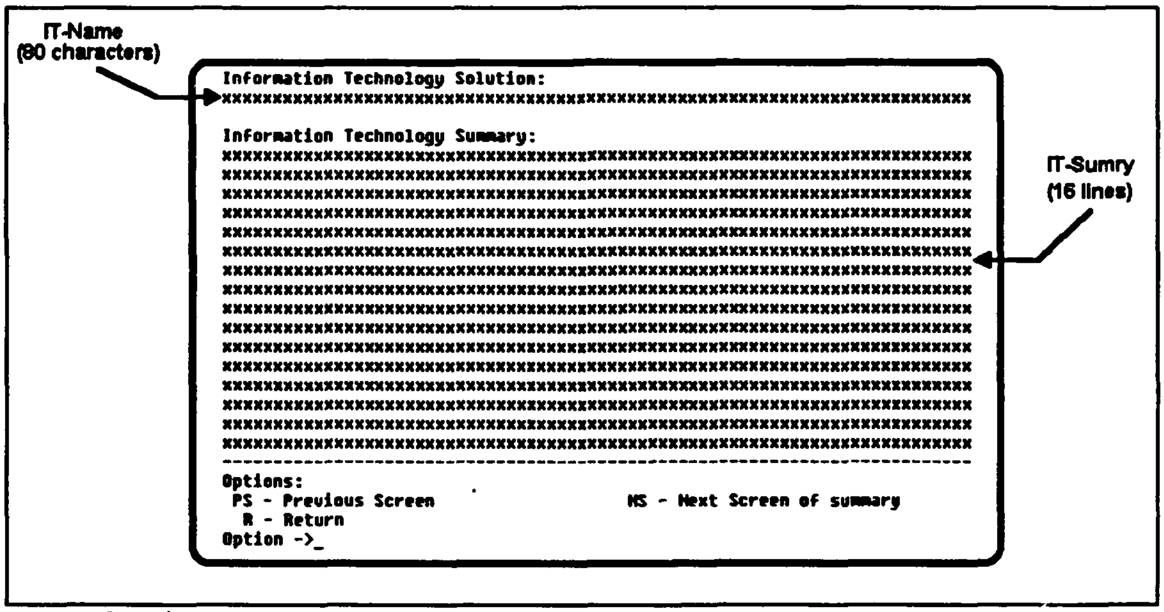
Benchmark Summary screen 1



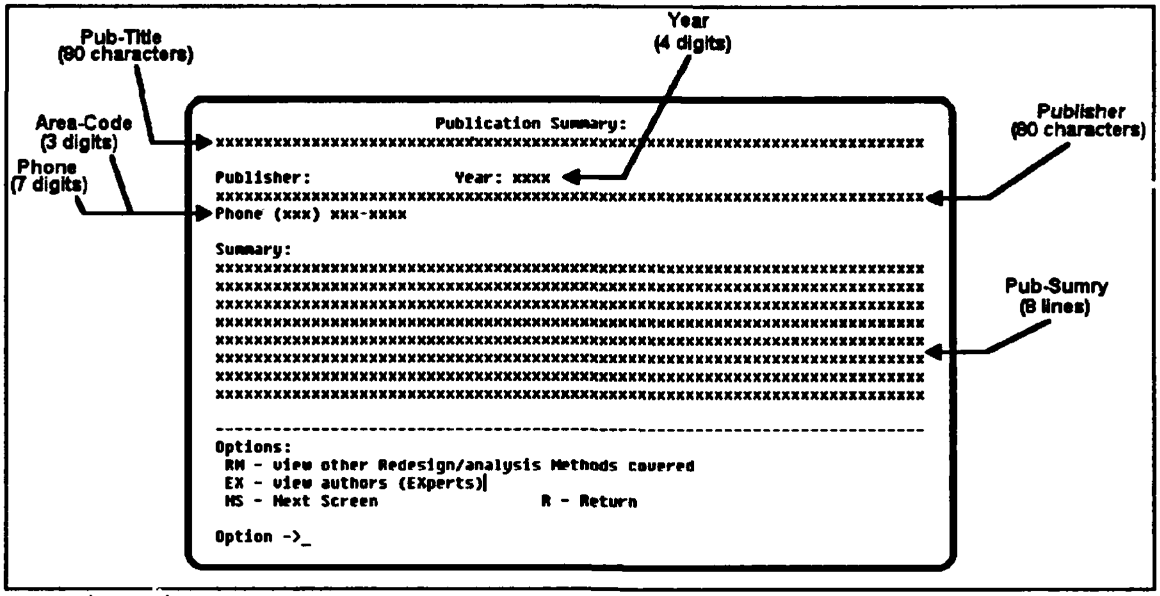
Benchmark Summary screen 2



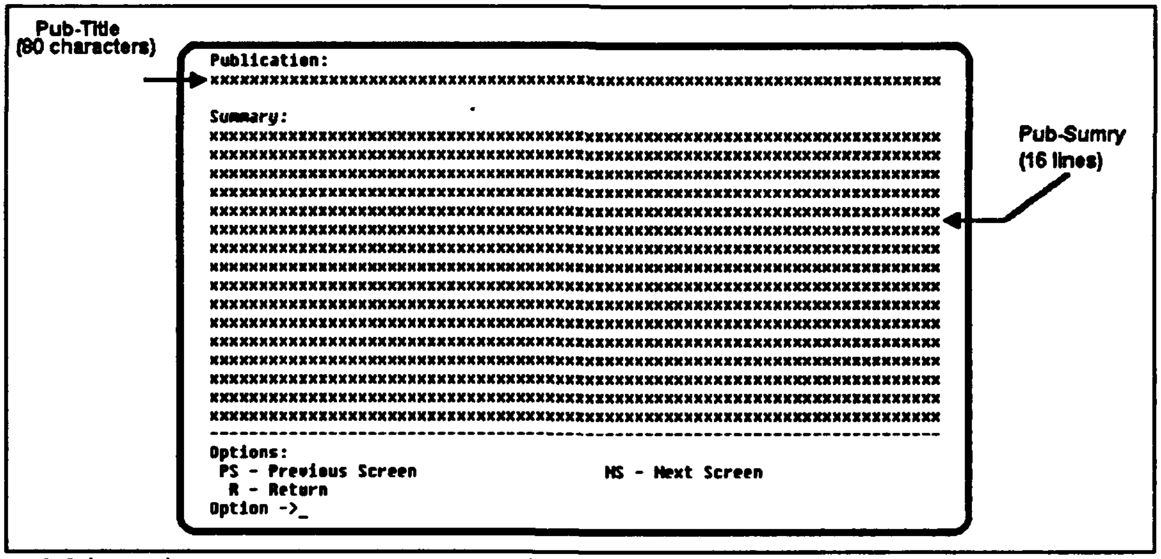
IT Solution Summary screen 1



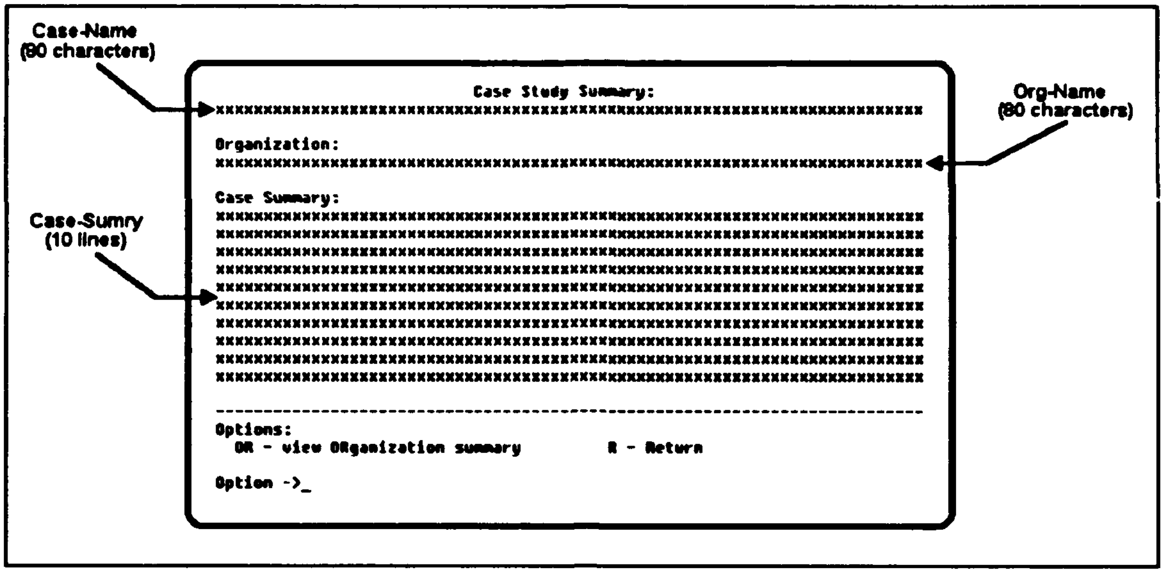
IT Solution Summary screen 2



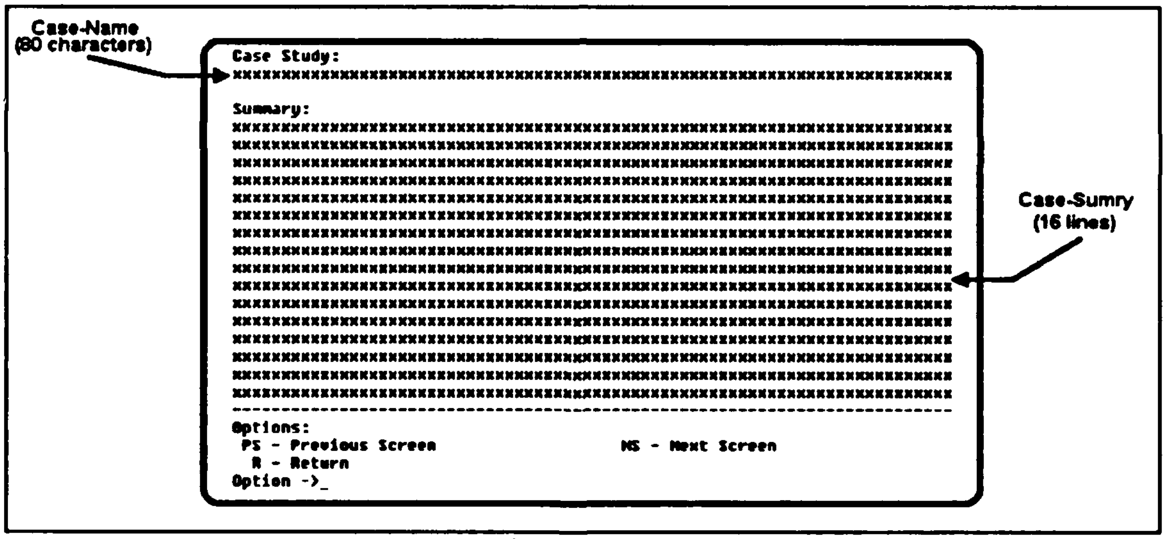
Publication Summary screen 1



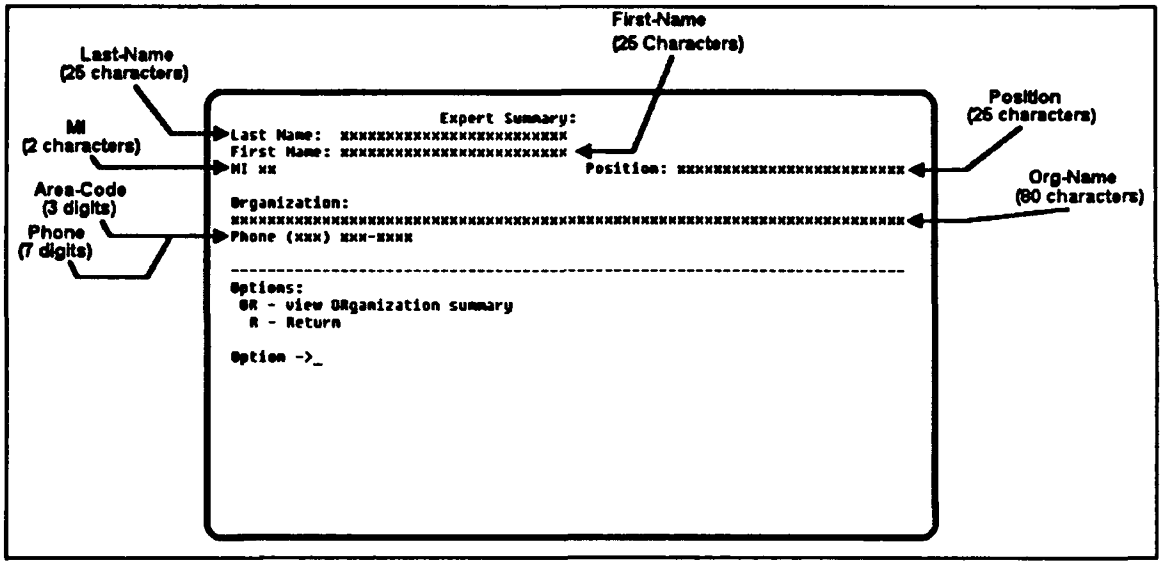
Publication Summary screen 2



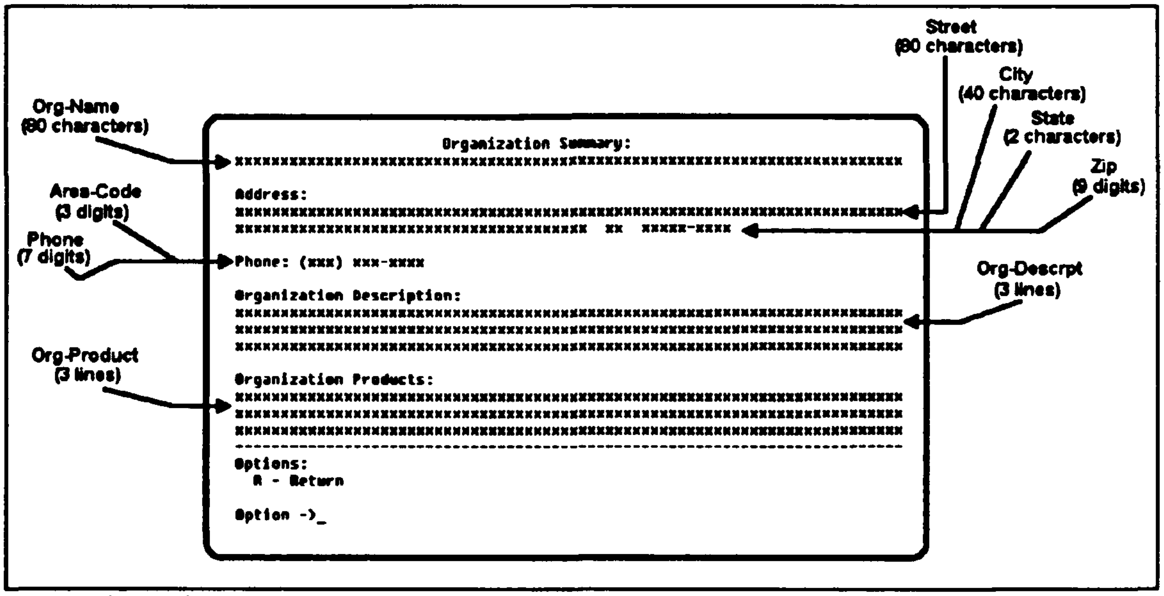
Case Study Summary screen 1



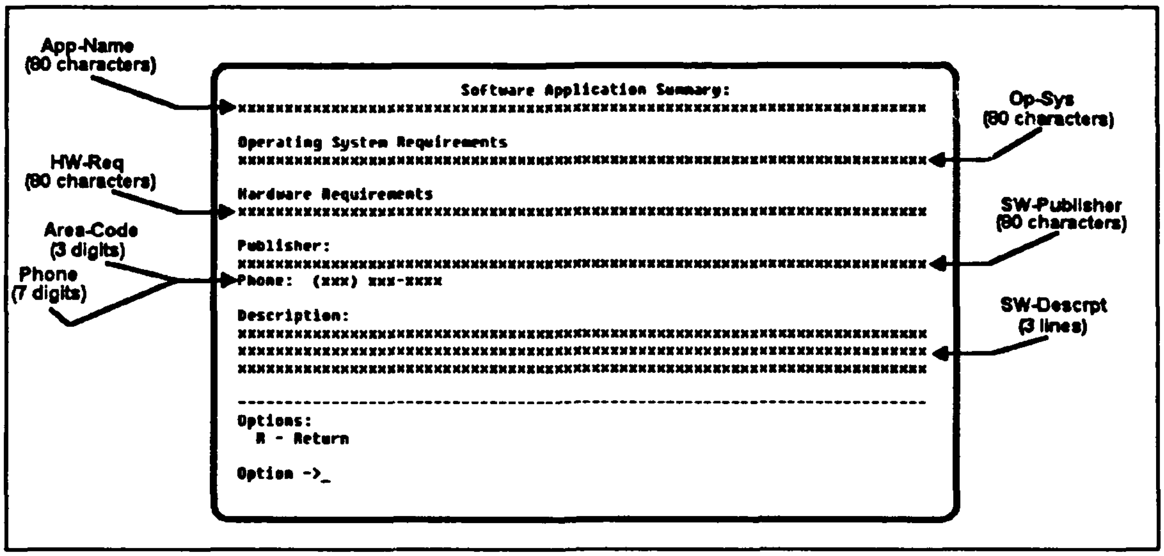
Case Study Summary screen 2



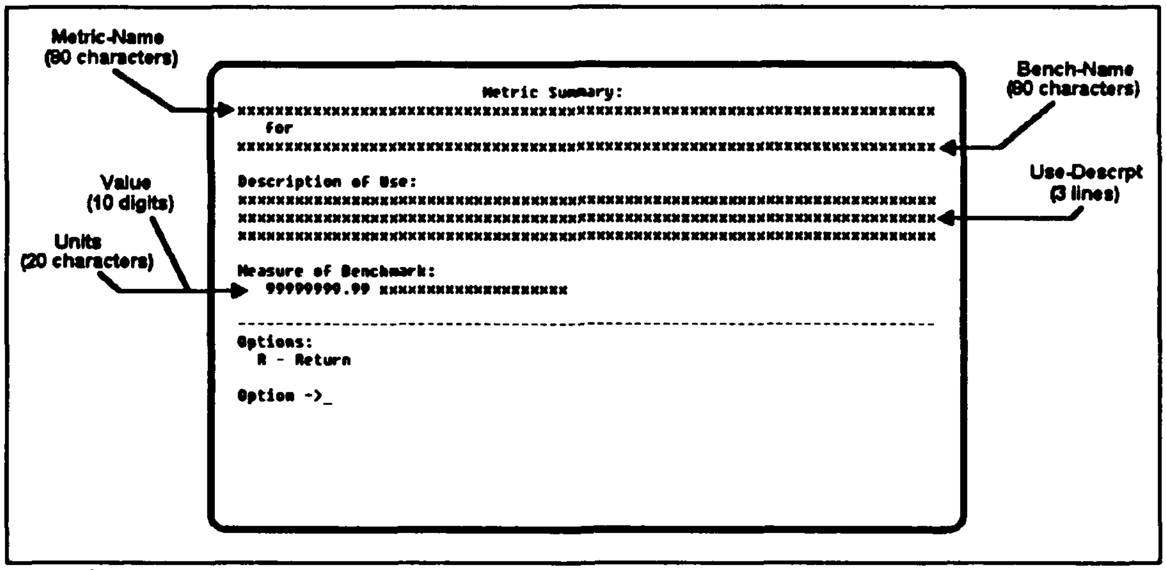
Expert Summary screen



Organization Summary screen



Software Summary screen



Metric Summary screen

APPENDIX F. ORACLE TABLES FOR THE REAP DATABASE PROTOTYPE

```
     AREA;
Name                             Null?    Type
-------------------------------  -------- ----
AREANAME                                  CHAR(80)
ADESCRPT                                  CHAR(240)
```

```
     METHODS;
Name                             Null?    Type
-------------------------------  -------- ----
METHNAME                                  CHAR(80)
MSUMRY                                    CHAR(80)
MRESULT                                   CHAR(160)
```

```
     BENCHMARK;
Name                             Null?    Type
-------------------------------  -------- ----
BENCHNAME                                 CHAR(80)
PROCSUMRY                                 CHAR(240)
ORGNAME                                   CHAR(80)
CASENAME                                  CHAR(80)
AREANAME                                  CHAR(80)
```

```
     ITSOLUTION;
Name                             Null?    Type
-------------------------------  -------- ----
ITNAME                                    CHAR(80)
ITSUMRY                                   CHAR(80)
SYSREQ                                    CHAR(80)
IMPACT                                    CHAR(160)
```

```
     CASESTUDY;
Name                             Null?    Type
-------------------------------  -------- ----
CASENAME                                  CHAR(80)
CASESUMRY                                 CHAR(160)
ORGNAME                                   CHAR(80)
```

```
     EXPERT;
Name                            Null?    Type
------------------------------- -------- ----
LASTNAME                                 CHAR(25)
FIRSTNAME                                CHAR(25)
MI                                       CHAR(2)
POSITION                                 CHAR(25)
AREACODE                                 CHAR(3)
PHONE                                    CHAR(7)
ORGNAME                                  CHAR(80)
```

```
     ORGANIZ;
Name                            Null?    Type
------------------------------- -------- ----
ORGNAME                                  CHAR(80)
STREET                                   CHAR(80)
CITY                                     CHAR(40)
STATE                                    CHAR(2)
ZIP                                      CHAR(9)
AREACODE                                 CHAR(3)
PHONE                                    CHAR(7)
ORGPRODUCT                               CHAR(160)
ORGDESCRPT                               CHAR(160)
```

```
     PUBLICATN;
Name                            Null?    Type
------------------------------- -------- ----
PUBTITLE                                 CHAR(80)
PUBLISHER                                CHAR(40)
AREACODE                                 CHAR(3)
PHONE                                    CHAR(7)
YEAR                                     CHAR(4)
PUBSUMRY                                 CHAR(80)
```

```
     SOFTWARE;
Name                            Null?    Type
------------------------------- -------- ----
APPNAME                                  CHAR(80)
OPSYS                                    CHAR(80)
HWREQ                                    CHAR(80)
SWDESCRPT                                CHAR(160)
SWPUBLISHER                              CHAR(40)
AREACODE                                 CHAR(3)
PHONE                                    CHAR(7)
```

```
     METRIC;
Name                              Null?    Type
------------------------------- -------- ----
METRICNAME                                 CHAR(80)
USEDESCRPT                                 CHAR(160)
METRICUNITS                                CHAR(40)
```

```
     AREA_METHOD;
Name                              Null?    Type
------------------------------- -------- ----
AREANAME                                   CHAR(80)
METHNAME                                   CHAR(80)
```

```
     AREA_ITSOLUTION;
Name                              Null?    Type
------------------------------- -------- ----
AREANAME                                   CHAR(80)
ITNAME                                     CHAR(80)
```

```
     BENCH_METRIC;
Name                              Null?    Type
------------------------------- -------- ----
BENCHNAME                                  CHAR(80)
METRICNAME                                 CHAR(80)
BENCHVALUE                                 NUMBER(8,2)
```

```
     ITSOL_CASE;
Name                              Null?    Type
------------------------------- -------- ----
ITNAME                                     CHAR(80)
CASENAME                                   CHAR(80)
```

```
     ITSOL_SW;
Name                              Null?    Type
------------------------------- -------- ----
ITNAME                                     CHAR(80)
APPNAME                                    CHAR(80)
```

```
     METHOD_EXPERT;
Name                            Null?    Type
------------------------------- -------- ----
METHNAME                                 CHAR(80)
LASTNAME                                 CHAR(25)
FIRSTNAME                                CHAR(25)
MI                                       CHAR(2)
```

```
     METHOD_SW;
Name                            Null?    Type
------------------------------- -------- ----
METHNAME                                 CHAR(80)
APPNAME                                  CHAR(80)
```

```
     METHOD_ORG;
Name                            Null?    Type
------------------------------- -------- ----
METHNAME                                 CHAR(80)
ORGNAME                                  CHAR(80)
```

```
     METHOD_PUB;
Name                            Null?    Type
------------------------------- -------- ----
METHNAME                                 CHAR(80)
PUBTITLE                                 CHAR(80)
```

```
     METHOD_CASE;
Name                            Null?    Type
------------------------------- -------- ----
METHNAME                                 CHAR(80)
CASENAME                                 CHAR(80)
```

```
     PUB_EXPERT;
Name                            Null?    Type
------------------------------- -------- ----
PUBTITLE                                 CHAR(80)
LASTNAME                                 CHAR(25)
FIRSTNAME                                CHAR(25)
MI                                       CHAR(2)
```

APPENDIX G. TEST QUERIES AND RESULTS

```
TEST 1 - LIST OF AREAS
  SQL> run
  1  select areaname
  2* from area

AREANAME
--------------------------------------------------------------------------------
Civilian Payroll
Travel
Retired Pay
Contract Payment
Financial Operations
Government Furnished Materials
Civilian Pernonnel
Depot Maintenance
Materials Requirements
Distribution Center Operations

10 records selected.

TEST 2 - AREA SUMMARY
SQL> run
  1  select *
  2  from area
  3* where areaname = 'Distribution Center Operations'

AREANAME
--------------------------------------------------------------------------------
ADESCRPT
--------------------------------------------------------------------------------

Distribution Center Operations
All activites associated with the control and management of logistics distribut
ion centers in DOD.
```

```
TEST 3 - METHODS RELATED TO AN AREA
SQL> run
  1  select methname
  2  from area_method
  3* where areaname = 'Distribution Center Operations'

METHNAME
--------------------------------------------------------------------------------
Activity Based Costing
Benchmarking
Painting the Bridge
Total Quality Management

TEST 4 - METHOD SUMMARY
SQL> run
  1  select *
  2  from methods
  3* where methname = 'Activity Based Costing'

METHNAME
--------------------------------------------------------------------------------
MSUMRY
--------------------------------------------------------------------------------
MRESULT
--------------------------------------------------------------------------------
Activity Based Costing
Activity Based Costing (ABC) attempts to assign costs to the activities ........

A more accurate view of the cost drivers in a process.  Identification of non-v
alue added activities.

TEST 6 - BENCHMARKS OF A SPECIFIC AREA
SQL> run
  1  select benchname
  2  from benchmark
  3* where areaname = 'Distribution Center Operations'

BENCHNAME
--------------------------------------------------------------------------------
L.L. Bean Distribution System

TEST 7 - BENCHMARK SUMMARY
SQL> run
  1  select *
  2  from benchmark
  3* where benchname = 'L.L. Bean Distribution System'

BENCHNAME
--------------------------------------------------------------------------------
```

```
PROCSUMRY
--------------------------------------------------------------------------------
ORGNAME
--------------------------------------------------------------------------------
CASENAME
--------------------------------------------------------------------------------
AREANAME
--------------------------------------------------------------------------------
L.L. Bean Distribution System
Process summary of LL Bean Distribution System.  Customer orrers are processed
by...
L.L. Bean
L.L. Bean Distribution Operations
Distribution Center Operations
```

TEST 8 - IT SOLUTIONS RELATED TO A SPECIFIC AREA

```
SQL> run
  1  select itname
  2  from area_itsolution
  3* where areaname = 'Distribution Center Operations'

ITNAME
--------------------------------------------------------------------------------
Expert Systems
Document Imaging
Local Area Networks
```

TEST 9 - IT SOLUTION SUMMARY

```
SQL> run
  1  select *
  2  from itsolution
  3* where itname = 'Expert Systems'

ITNAME
--------------------------------------------------------------------------------
ITSUMRY
--------------------------------------------------------------------------------
SYSREQ
--------------------------------------------------------------------------------
IMPACT
--------------------------------------------------------------------------------
Expert Systems                                                               An
expert system consists of a knowledge base and a inference engine...         A
PC/Work station, Expert System software, an Expert or Knowledge base     Rapid
diognosis or problen\m solving.  More consistent decisions
```

TEST 10 - CASE STUDIES RELATED TO A METHOD

```
SQL> run
  1  select casename
  2  from method_case
  3* where methname = 'Business Process Reengineering'

CASENAME
--------------------------------------------------------------------------------
US Army Corps of Engineers Business Reengineering
Boeing Develops New Design and Manfacturing Team
Ford Motor Company Accounts Payable
Tactical Air Command Decentralization
```

TEST 11 - PUBLICATIONS RELATED TO A METHOD

```
SQL> run
  1  select pubtitle
  2  from method_pub
  3* where methname = 'Activity Based Costing'

PUBTITLE
--------------------------------------------------------------------------------
Activity Accounting: An Activity-Based Costing Approach
Common Cents: The ABC Performance Breakthrough
Cost management at Boeing Helicopter
Theory of Constraints vs. ABC:  Is there one best solution?
Decision Based Costing, Generalized ABC?
Cost Defined by Responsibilities                                             The
ABC Evolution at Rockwell
Activity Based Information As the Foundation of World Class Performance
Driving in ABCA while implementing TQM
Management Accounting  2nd Ed.                                               The
Performance Effects of ABC and ABM systems                                Profit
Priorities from ABC                                                    Designing
and Implementing a New Cost Management System                        Costing for
Wharehousing and Distribution                                     Activity-Based
Costing for Marketing                                         Are You Distorting
Costs by Violating ABC Assumptions?                                 Elgin Sweeper
Company's Journey Toward Cost Management
17 records selected.
```

TEST 12 - OTHER METHODS COVERED IN A SPECIFIC PUBLICATION

```
SQL> run
  1  select methname
  2  from Method_pub
  3* where pubtitle = 'Driving in ABCA while implementing TQM'

METHNAME
--------------------------------------------------------------------------------
Activity Based Costing
Total Quality Management
```

```
TEST 13 - PUBLICATION SUMMARY
SQL> run
  1  select *
  2  from publicatn
  3* where pubtitle = 'Driving in ABCA while implementing TQM'

PUBTITLE
--------------------------------------------------------------------------------
PUBLISHER                                     ARE PHONE   YEAR
--------------------------------------------- --- ------- ----
PUBSUMRY
--------------------------------------------------------------------------------
Driving in ABCA while implementing TQM
CAMI/CMS NAVAIR Depot Cherry Point            817 8601654 1991
A brief presented by the at the December 91 CAM-I conference. Focus on ........

TEST 14 - EXPERTS RELATED TO A METHOD
SQL> run
  1  select lastname, firstname, mi
  2  from method_expert
  3* where methname = 'Benchmarking'

LASTNAME                  FIRSTNAME                 MI
------------------------- ------------------------- --
Yearout                   Stephen                   L
Hammond                   Joshua

TEST 15 - EXPERT SUMMARY
SQL> run
  1  select *
  2  from expert
  3  where lastname = 'Yearout'
  4  and firstname = 'Stephen'
  5* and mi = 'L'

LASTNAME                  FIRSTNAME                 MI
------------------------- ------------------------- --
POSITION                  ARE PHONE
------------------------- --- -------
ORGNAME
--------------------------------------------------------------------------------
Yearout                   Stephen                   L
Natl Dir Ops&Quailty Mgmt 216 8615000
Ernst & Young
```

```
TEST 16 - ORGANIZATIONS RELATED TO A METHOD
SQL> run
  1  select orgname
  2  from method_org
  3* where methname = 'Benchmarking'

ORGNAME
------------------------------------------------------------------------------
Real Decisions Corporation
Ernst & Young
American Quality Foundation
International Benchmarking Clearinghouse - APQC

TEST 17 - SOFTWARE RELATED TO A METHOD
SQL> run
  1  select appname
  2  from method_sw
  3* where methname = 'Activity Based Costing'

APPNAME
------------------------------------------------------------------------------
Easy ABC Quick
Alpha Cost
Profit Manager Junior

TEST 18 - CASE STUDIES RELATED TO AN IT SOLUTION
SQL> run
  1  select casename
  2  from itsol_case
  3* where itname = 'Document Imaging'

CASENAME
------------------------------------------------------------------------------
USAAs Automation Processes

TEST 19 - SOFTWARE RELATED TO AN IT SOLUTION
SQL> run
  1  select appname
  2  from itsol_sw
  3* where itname = 'Decision Support Systems'

APPNAME
------------------------------------------------------------------------------
Quattro Pro 4.0
Lotus 123
IFPS
```

```
TEST 20 - METRICS USED TO MEASURE A BENCHMARK
SQL> run
  1  select metricname
  2  from bench_metric
  3* where benchname = 'L.L. Bean Distribution System'

METRICNAME
--------------------------------------------------------------------------------
Number of Orders per man-day
Number of Pieces per man-day
Number of Lines per man-day
Order turn around time

TEST 21 - BENCHMARK MEASURE SUMMARY *****(NOTE: 2 TABLES USED)*****
SQL> run
  1  select bench_metric.benchname, bench_metric.benchvalue, metric.metricunits,
  2  metric.usedescrpt
  3  from bench_metric, metric
  4  where bench_metric.benchname = 'L.L. Bean Distribution System'
  5  and bench_metric.metricname = 'Number of Lines per man-day'
  6* and metric.metricname = bench_metric.metricname

BENCHNAME
--------------------------------------------------------------------------------
BENCHVALUE METRICUNITS                                                          -
--------- ----------------------------------------
USEDESCRPT
--------------------------------------------------------------------------------
L.L. Bean Distribution System
      1440 lines/man-day
Used to measure the number of trips from a point in the wharehouse to the bin.

TEST 22 - BENCHMARK ORGANIZATION SUMMARY
SQL> run
  1  select benchmark.benchname, organiz.*
  2  from benchmark, organiz
  3  where benchmark.benchname = 'L.L. Bean Distribution System'
  4* and benchmark.orgname = organiz.orgname

BENCHNAME
--------------------------------------------------------------------------------
ORGNAME
--------------------------------------------------------------------------------
STREET
--------------------------------------------------------------------------------
CITY                                     ST ZIP       ARE PHONE
---------------------------------------- -- --------- --- -------
ORGPRODUCT
--------------------------------------------------------------------------------
```

```
ORGDESCRPT
--------------------------------------------------------------------------------

L.L. Bean Distribution System
L.L. Bean
123 Main St.
Freeport                                    MN 001231000 800 5551212
Outdoor clothing and equipment
L.L. Bean is a catalog/phone order outdoor clother.

TEST 23 - BENCHMARK CASE STUDY SUMMARY
SQL> run
  1  select benchmark.benchname, casestudy.*
  2  from benchmark, casestudy
  3  where benchmark.benchname = 'L.L. Bean Distribution System'
  4* and benchmark.casename = casestudy.casename

BENCHNAME
--------------------------------------------------------------------------------
CASENAME
--------------------------------------------------------------------------------
CASESUMRY
--------------------------------------------------------------------------------
ORGNAME
--------------------------------------------------------------------------------
L.L. Bean Distribution System
L.L. Bean Distribution Operations
L.L. Bean maintains its nation-wide order center and supporting wharehouse oope
rations together in Freeport Maine. While manually intensive, its wharehouse...
L.L. Bean

TEST 24 - ORGANIZATION RELATED TO CASE STUDY SUMMARY
SQL> run
  1  select casestudy.casename, organiz.*
  2  from casestudy, organiz
  3  where casestudy.casename = 'Norfolk Naval Shipyard Implements TQM'
  4* and casestudy.orgname = organiz.orgname

CASENAME
--------------------------------------------------------------------------------
ORGNAME
--------------------------------------------------------------------------------
STREET
--------------------------------------------------------------------------------
CITY                                        ST ZIP         ARE PHONE
---------------------------------------- -- --------- --- -------
ORGPRODUCT
--------------------------------------------------------------------------------
ORGDESCRPT
--------------------------------------------------------------------------------
```

```
Norfolk Naval Shipyard Implements TQM
USN Norfolk Naval Shipyard
Norfolk Naval Shipyard
Norfolk                                   VA 200000    408 5551212
Overhaul of USN ships to include combatants, auxillaries and submarines.  Conve
ntial and nuclear powered units.
Largest USN shipyard.  Located adjacent to Norfolk Navalbase.

TEST 25 - EXPERT AND RELATED ORGANIZATION SUMMARY
SQL> run
  1  select expert.*, organiz.*
  2  from expert, organiz
  3  where expert.lastname = 'Yearout'
  4  and expert.firstname = 'Stephen'
  5  and expert.mi = 'L'
  6* and expert.orgname = organiz.orgname

LASTNAME                   FIRSTNAME                  MI
-------------------------- -------------------------- --
POSITION                   ARE PHONE
-------------------------- --- -------
ORGNAME
--------------------------------------------------------------------------------
ORGNAME
--------------------------------------------------------------------------------
STREET
--------------------------------------------------------------------------------
CITY                                         ST ZIP        ARE PHONE
-------------------------------------------- -- ---------- --- -------
ORGPRODUCT
--------------------------------------------------------------------------------
ORGDESCRPT
--------------------------------------------------------------------------------
Yearout                    Stephen                    L
Natl Dir Ops&Quailty Mgmt  216 8615000
Ernst & Young
Ernst & Young
1600 Huntington Building
Cleveland                                    OH 44115      216 8615000
Benchmark comparison
In conjunction with AQF, developed the International Quality Study database (hu
ge). Conducts benchmark evaluations against the database.
```

```
TEST 26 - AUTHOR(S) OF A PUBLICATION
SQL> run
  1  select lastname, firstname, mi
  2  from pub_expert
  3* where pubtitle = 'Profit Priorities from ABC'

LASTNAME                   FIRSTNAME                  MI
-------------------------  -------------------------  --
Cooper                     R
Kaplan                     R

TEST 27 - SOFTWARE SUMMARY
SQL> run
  1  select *
  2  from software
  3* where appname = 'Quattro Pro 4.0'

APPNAME
--------------------------------------------------------------------------------
OPSYS
--------------------------------------------------------------------------------
HWREQ
--------------------------------------------------------------------------------
SWDESCRPT
--------------------------------------------------------------------------------
SWPUBLISHER                                 ARE PHONE
------------------------------------------  --- -------
Quattro Pro 4.0
MS-DOS, Windows 3.x                                                          IBM
compatible PC, 512k RAM, 5 Meg on harddisk
Electronic spreadsheet.  Graphics capability.
Borland International                       408 4388400
```

LIST OF REFERENCES

Boehm, B.W., *Software Risk Management: Principles and Practices*,
1989 (Video Notes) IEEE Computer Society Press.

Brimson, J.A., *Activity Accounting: An Activity-Based Costing Approach*,
1991 John Wiley & Sons Inc., New York.

Brooks, F, *The Mythical Man-Month*,
1975 Addison Wesley, Reading Massachusetts.

Camp, R.C., *Benchmarking, The Search for Industry Best Practices that lead to*
1989 *Superior Performance*, American Society for Quality Control (ASQC) Quality Press, Milwaukee, Wisconsin.

Card, D., "Understanding Process Improvement"
1991 *IEEE Software*, July 1991, pp 102-103.

Cooper, R. and Kaplin, R.S., "Profit Priorities from Activity-Based Costing"
1991 *Harvard Business Review*, May-June 1991.

Department of Administrative Sciences, Naval Postgraduate School
Feb, 1992 Letter to Michael Yoemans, Office of the Director of Defense Information, Subject: Agreement Regarding Tasks to Support the Office of the Director of Defense Information (U), 12 February 1992.

Dimmick, S., et al. *The Oracle Database Administrator's Guide Version 5.1*,
1989 Oracle Corporation, Belmont CA.

Harrington, J., Business Process Improvement: *The breakthrough strategy for*
1991 *total quality, productivity and competitiveness*, McGraw Hill Inc., New York.

Hulme, C., "Reading: Extracting Information from Printed and Electronically
1984 Presented Text", Chapter 2 of *Fundamentals of Human-Computer Interaction*, Edited by Monk, A., Academic Press, London.

Kroenke, D.M., and Dolan, K.A. *Database Processing*, 3d ed.,
1988 Macmillan Publishing Company, New York.

Network Information Center, *Query of information* database, Defense Data Network-
Aug 1992 Network Information Center, Government Systems Inc.

Page-Jones,M., *The Practical Guide to Structured System Design 2 ed.*,
1988 Prentice-Hall, Inc., Englewood Cliffs, NJ.

Postel, J., and J. Reynolds, "Telnet Protocol Specification", *RFC 854*,
1983 USC Information Sciences Institute, May 1983.

Pressman, R.S., *Software Engineering: A Practitioner's Approach*,
1992 3d ed., McGraw Hill, New York.

Schlaer, S. and Mellor, S., *Object Oriented System Analysis*,
1988 Yourdon Press.

Shores, R.A., *A TQM Approach to Achieving Manufacturing Excellence*,
1990 American Society for Quality Control (ASQC) Quality Press, Milwaukee, Wisconsin.

Strassmann, P.A., The DoD Context for I-CASE, presented at the I-CASE
1992 Bidders Conference, Montgomery, Alabama, March 31, 1992.

Spivey, D., Information Systems Modernization in the U.S. Army Corps of
1992 Engineers, presented at the Naval Post Graduate School on 4 June 1992, U.S. Army Corps of Engineers.

Teal,T., "Service Comes First: An interview with USAA's Robert F. McDermott",
1991 *Harvard Business Review*, September-October 1991.

Whitten J.L., Bentley L.D. and Barlow, V.M., *System Analysis and Design Methods*
1989 *2nd ed.*, Richard D. Irwin, Inc., Homewood, IL.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	William J. Haga Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5002	2
4.	Kenneth J. Euske Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5002	1